This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problems Mailbox.

US006540144B1

(12) United States Patent
Hudrick et al.

(10) Patent No.: US 6,540,144 B1
(45) Date of Patent: Apr. 1, 2003

(54) TECHNIQUES FOR INTERFACING A BAR CODE SCANNER TO A PC USING A KEYBOARD RETRANSMIT PROTOCOL

(75) Inventors: **Donald T. Hudrick**, Sicklerville, NJ (US); **John A. Furlong**, Woodbury, NJ (US); **Ian A. Scott**, Haddonfield, NJ (US)

(73) Assignee: **Metrologic Instruments, Inc.**, Blackwood, NJ (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 69 days.

(21) Appl. No.: **09/614,356**

(22) Filed: **Jul. 12, 2000**

Related U.S. Application Data

(60) Provisional application No. 60/144,361, filed on Jul. 16, 1999.

(51) **Int. Cl.**[7] **G06K 7/10**
(52) **U.S. Cl.** **235/462.15**
(58) **Field of Search** 235/462.15

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,021,640 A	* 6/1991	Muroi	....	186/61
5,178,234 A	1/1993	Sakurai et al.		
5,848,292 A	* 12/1998	Nathan	....	710/2
5,854,945 A	* 12/1998	Criscito et al.	....	235/435
5,875,415 A	2/1999	Lieb et al.		
6,263,383 B1	* 7/2001	Lee et al.	....	709/218
6,312,175 B1	* 11/2001	Lum	....	400/472

FOREIGN PATENT DOCUMENTS

JP 8-328717 * 12/1996

OTHER PUBLICATIONS

Symbol Technologies "ScanPlus Reference Manual" Edition 1.0 Apr., 1996, p. 4–39.*

Copyright 1996 Information Access Company; Automatic I.D. News; Sep., 1996; p. 18; ISSN: 0890–9768.

Copyright 1996 Information Access Company; Packaging Digest; May, 1996; vol. 33; No. 6; p. 34; ISSN: 0030–9117.

MS900 by Metrologic Instruments, Inc.; Sep. 1998; www.metrologic.com.

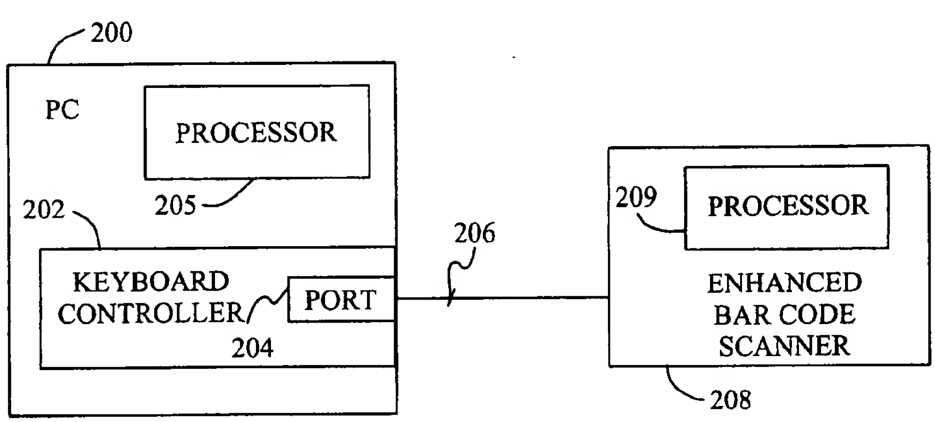
* cited by examiner

Primary Examiner—Mark Tremblay
(74) *Attorney, Agent, or Firm*—Morgan, Lewis & Bockius, LLP

(57) **ABSTRACT**

A bar code scanner is equipped with a keyboard emulation mechanism so as to provide an enhanced interface to a keyboard controller port of a computing device such as a personal or laptop computer. The keyboard emulation mechanism eliminates the necessity of connecting an actual computer keyboard to the port by responding to the computing device's standard power-on diagnostic procedure as if it were effectively an electrical equivalent of a computer keyboard. The keyboard emulation mechanism can also be equipped to detect a keyboard inhibit signal at the keyboard controller port. If the keyboard emulation mechanism detects a keyboard inhibit signal while a data byte is being transmitted to the keyboard controller port, the keyboard emulation mechanism retransmits this data byte to the keyboard controller port. The data byte may specify a scan code corresponding to a keyboard key press or all or part of a decoded bar code. This retransmission process is repeated up to a specified number of times, so as to provide additional opportunities for the scan code to be inputted to the keyboard controller port if the port is momentarily disabled by the keyboard inhibit signal. For many applications, it is advantageous to repeat the retransmission process up to three times for a given data byte. If the data byte is still not successfully received after the third attempt, the process is no longer repeated.

36 Claims, 13 Drawing Sheets



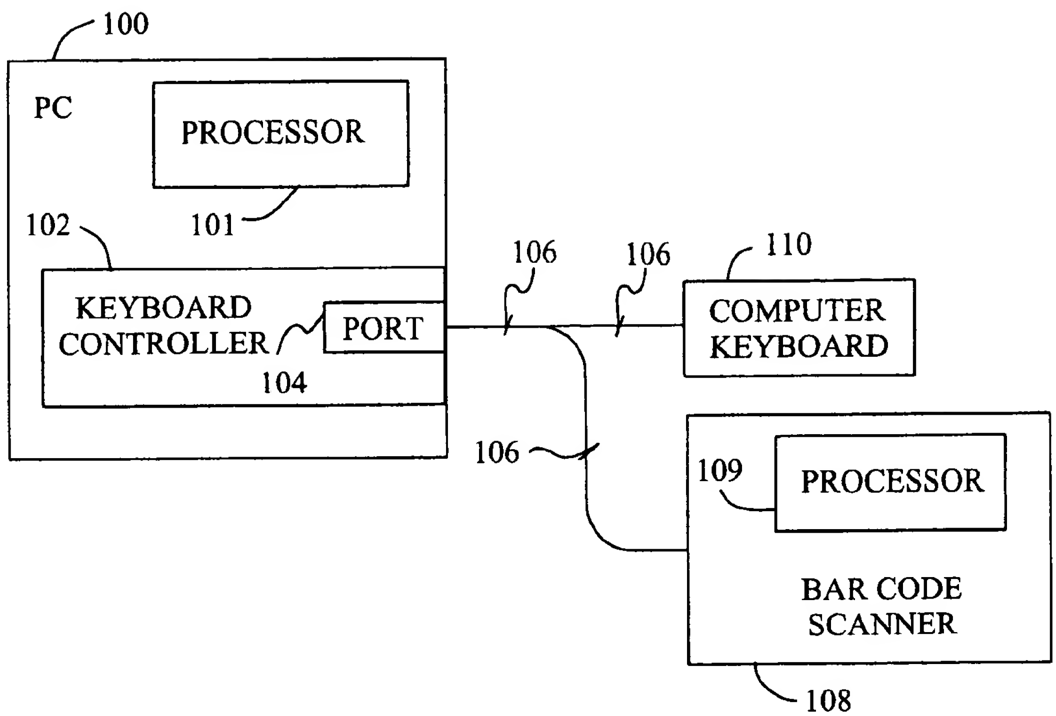


FIG. 1
PRIOR ART.

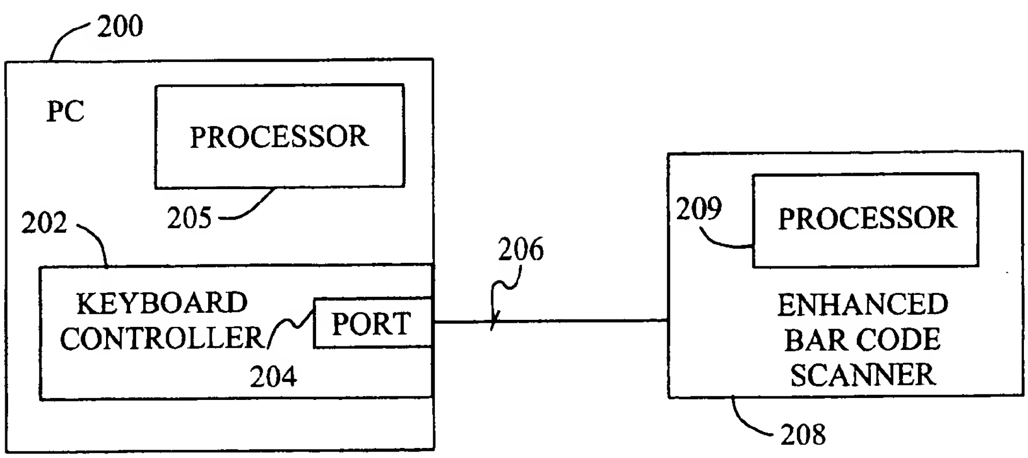

200
PC
PROCESSOR
205
202
KEYBOARD
CONTROLLER
PORT
204
206
209
PROCESSOR
ENHANCED
BAR CODE
SCANNER
208

FIG. 2

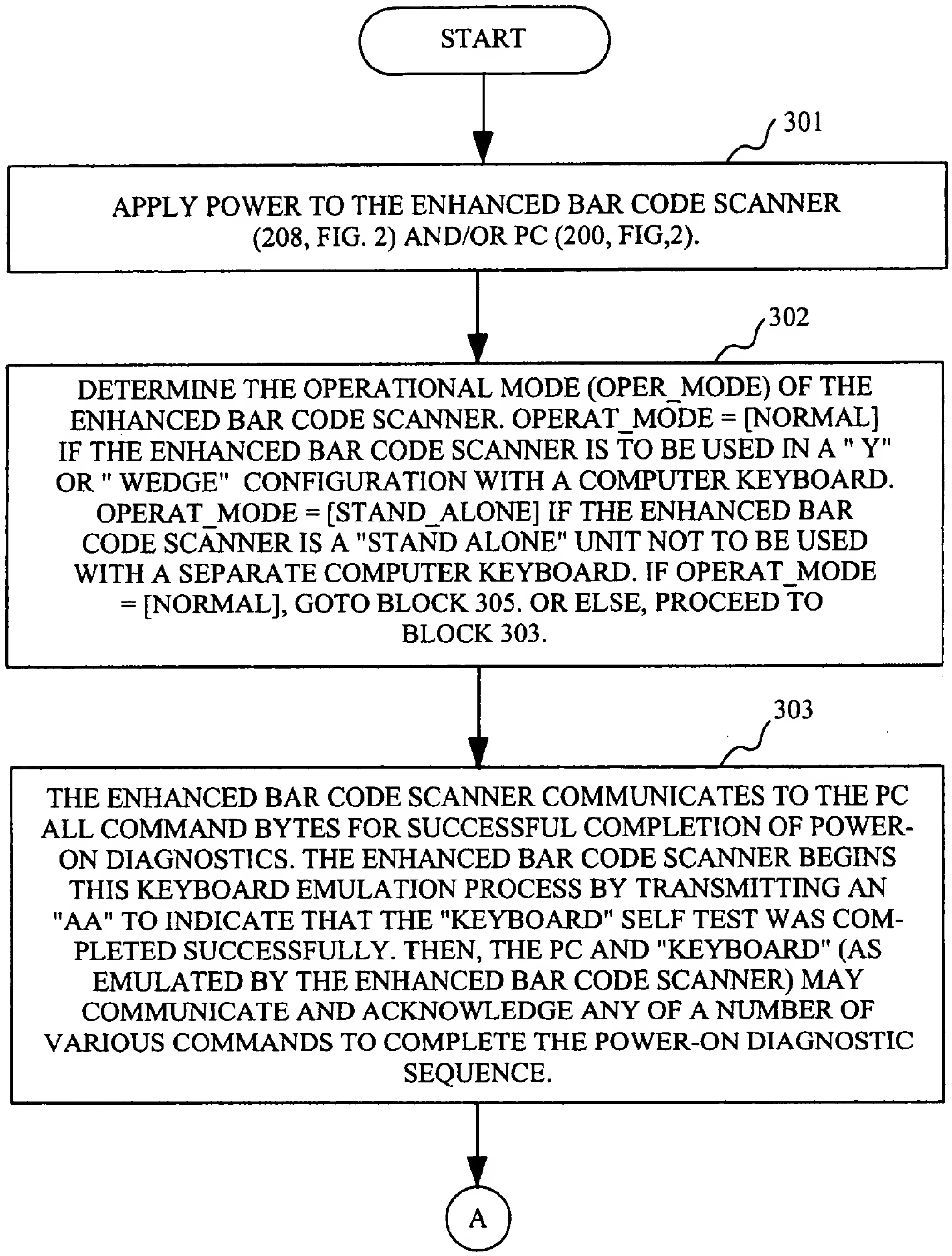

START
301
APPLY POWER TO THE ENHANCED BAR CODE SCANNER (208, FIG. 2) AND/OR PC (200, FIG,2).
302
DETERMINE THE OPERATIONAL MODE (OPER_MODE) OF THE ENHANCED BAR CODE SCANNER. OPERAT_MODE = [NORMAL] IF THE ENHANCED BAR CODE SCANNER IS TO BE USED IN A " Y" OR " WEDGE" CONFIGURATION WITH A COMPUTER KEYBOARD. OPERAT_MODE = [STAND_ALONE] IF THE ENHANCED BAR CODE SCANNER IS A "STAND ALONE" UNIT NOT TO BE USED WITH A SEPARATE COMPUTER KEYBOARD. IF OPERAT_MODE = [NORMAL], GOTO BLOCK 305. OR ELSE, PROCEED TO BLOCK 303.
303
THE ENHANCED BAR CODE SCANNER COMMUNICATES TO THE PC ALL COMMAND BYTES FOR SUCCESSFUL COMPLETION OF POWER-ON DIAGNOSTICS. THE ENHANCED BAR CODE SCANNER BEGINS THIS KEYBOARD EMULATION PROCESS BY TRANSMITTING AN "AA" TO INDICATE THAT THE "KEYBOARD" SELF TEST WAS COMPLETED SUCCESSFULLY. THEN, THE PC AND "KEYBOARD" (AS EMULATED BY THE ENHANCED BAR CODE SCANNER) MAY COMMUNICATE AND ACKNOWLEDGE ANY OF A NUMBER OF VARIOUS COMMANDS TO COMPLETE THE POWER-ON DIAGNOSTIC SEQUENCE.
A

FIG. 3A

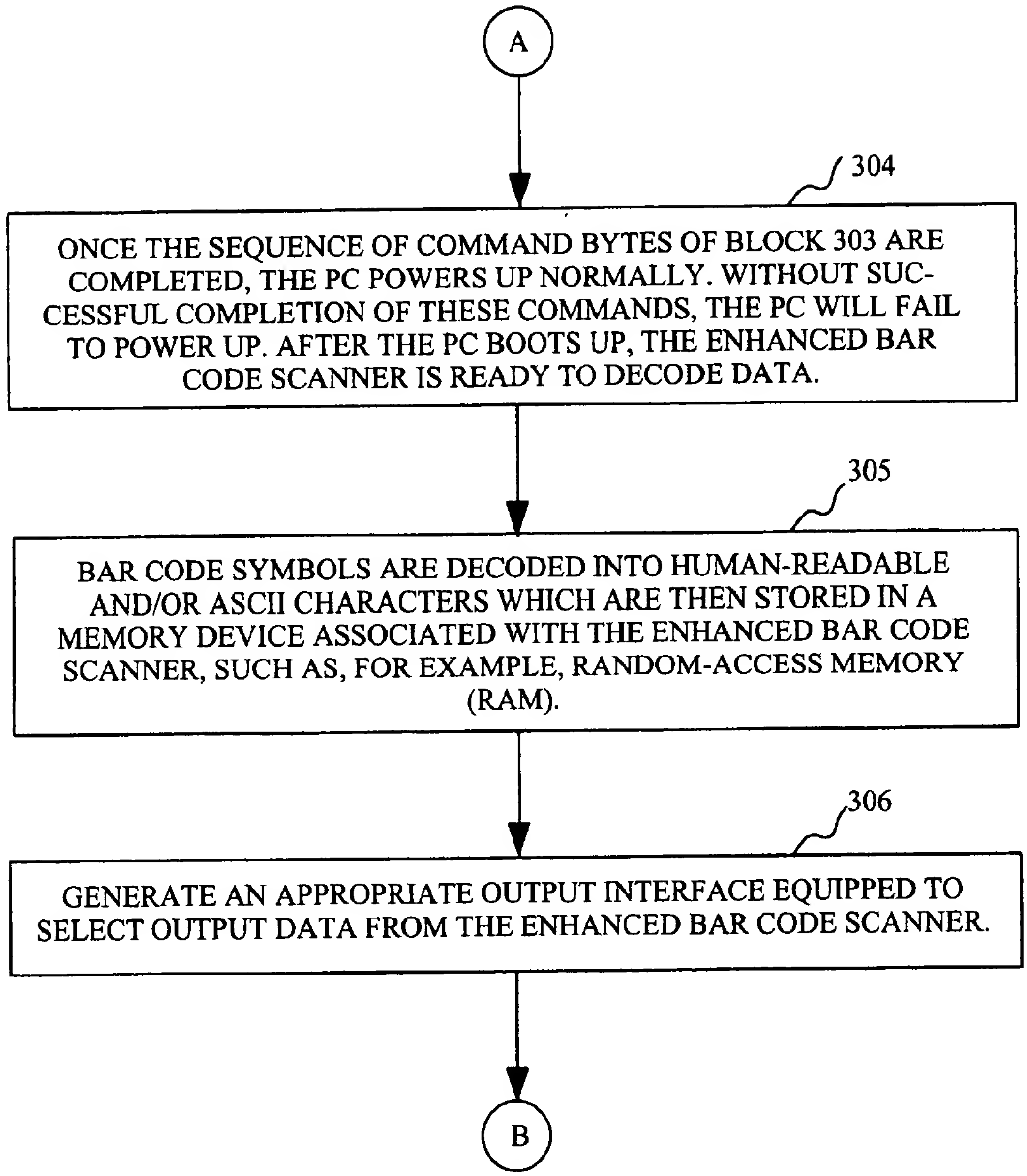

A
304
ONCE THE SEQUENCE OF COMMAND BYTES OF BLOCK 303 ARE COMPLETED, THE PC POWERS UP NORMALLY. WITHOUT SUCCESSFUL COMPLETION OF THESE COMMANDS, THE PC WILL FAIL TO POWER UP. AFTER THE PC BOOTS UP, THE ENHANCED BAR CODE SCANNER IS READY TO DECODE DATA.
305
BAR CODE SYMBOLS ARE DECODED INTO HUMAN-READABLE AND/OR ASCII CHARACTERS WHICH ARE THEN STORED IN A MEMORY DEVICE ASSOCIATED WITH THE ENHANCED BAR CODE SCANNER, SUCH AS, FOR EXAMPLE, RANDOM-ACCESS MEMORY (RAM).
306
GENERATE AN APPROPRIATE OUTPUT INTERFACE EQUIPPED TO SELECT OUTPUT DATA FROM THE ENHANCED BAR CODE SCANNER.
B

FIG. 3B

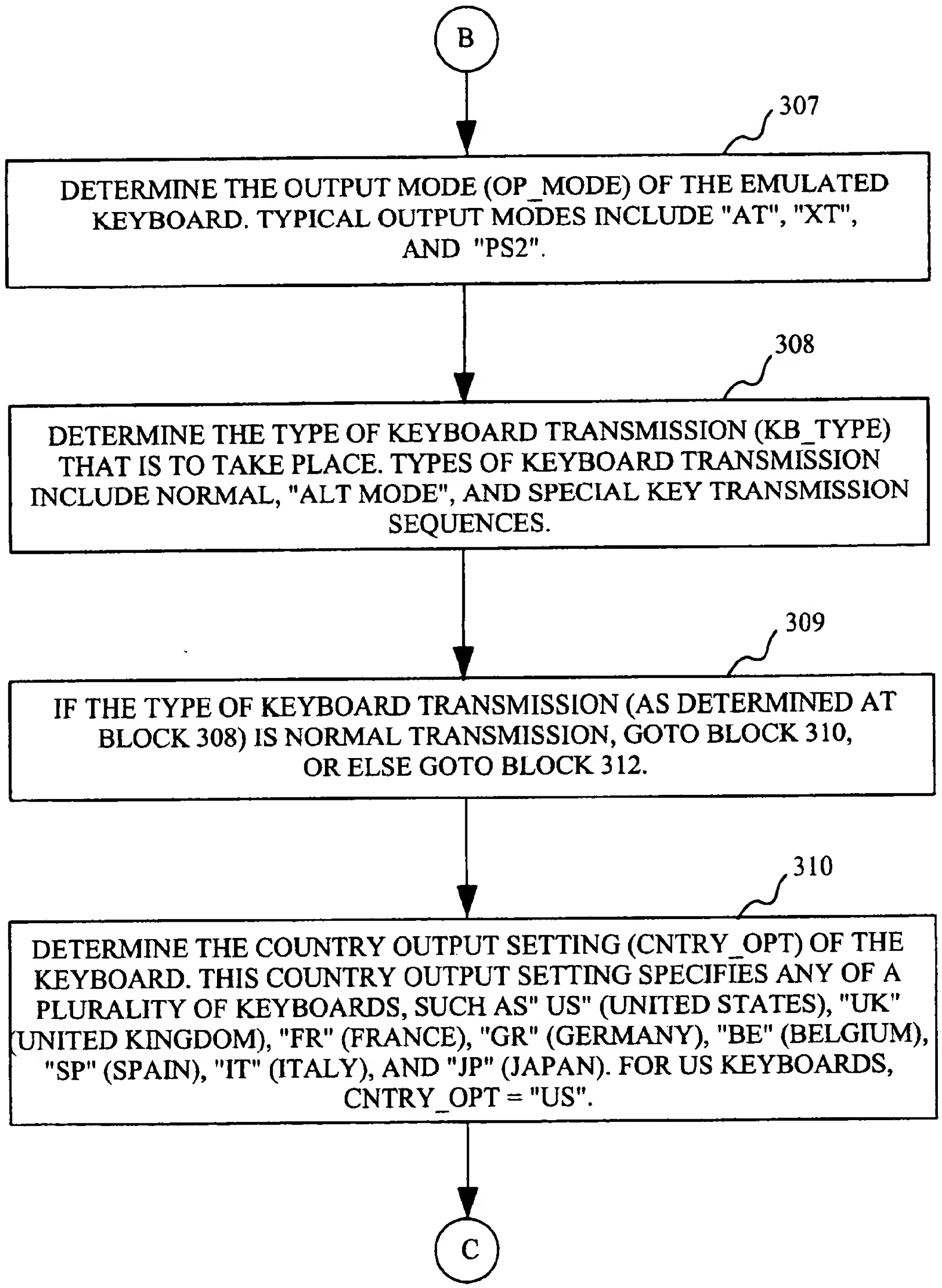

B
307
DETERMINE THE OUTPUT MODE (OP_MODE) OF THE EMULATED KEYBOARD. TYPICAL OUTPUT MODES INCLUDE "AT", "XT", AND "PS2".
308
DETERMINE THE TYPE OF KEYBOARD TRANSMISSION (KB_TYPE) THAT IS TO TAKE PLACE. TYPES OF KEYBOARD TRANSMISSION INCLUDE NORMAL, "ALT MODE", AND SPECIAL KEY TRANSMISSION SEQUENCES.
309
IF THE TYPE OF KEYBOARD TRANSMISSION (AS DETERMINED AT BLOCK 308) IS NORMAL TRANSMISSION, GOTO BLOCK 310, OR ELSE GOTO BLOCK 312.
310
DETERMINE THE COUNTRY OUTPUT SETTING (CNTRY_OPT) OF THE KEYBOARD. THIS COUNTRY OUTPUT SETTING SPECIFIES ANY OF A PLURALITY OF KEYBOARDS, SUCH AS" US" (UNITED STATES), "UK" (UNITED KINGDOM), "FR" (FRANCE), "GR" (GERMANY), "BE" (BELGIUM), "SP" (SPAIN), "IT" (ITALY), AND "JP" (JAPAN). FOR US KEYBOARDS, CNTRY_OPT = "US".
C

FIG. 3C

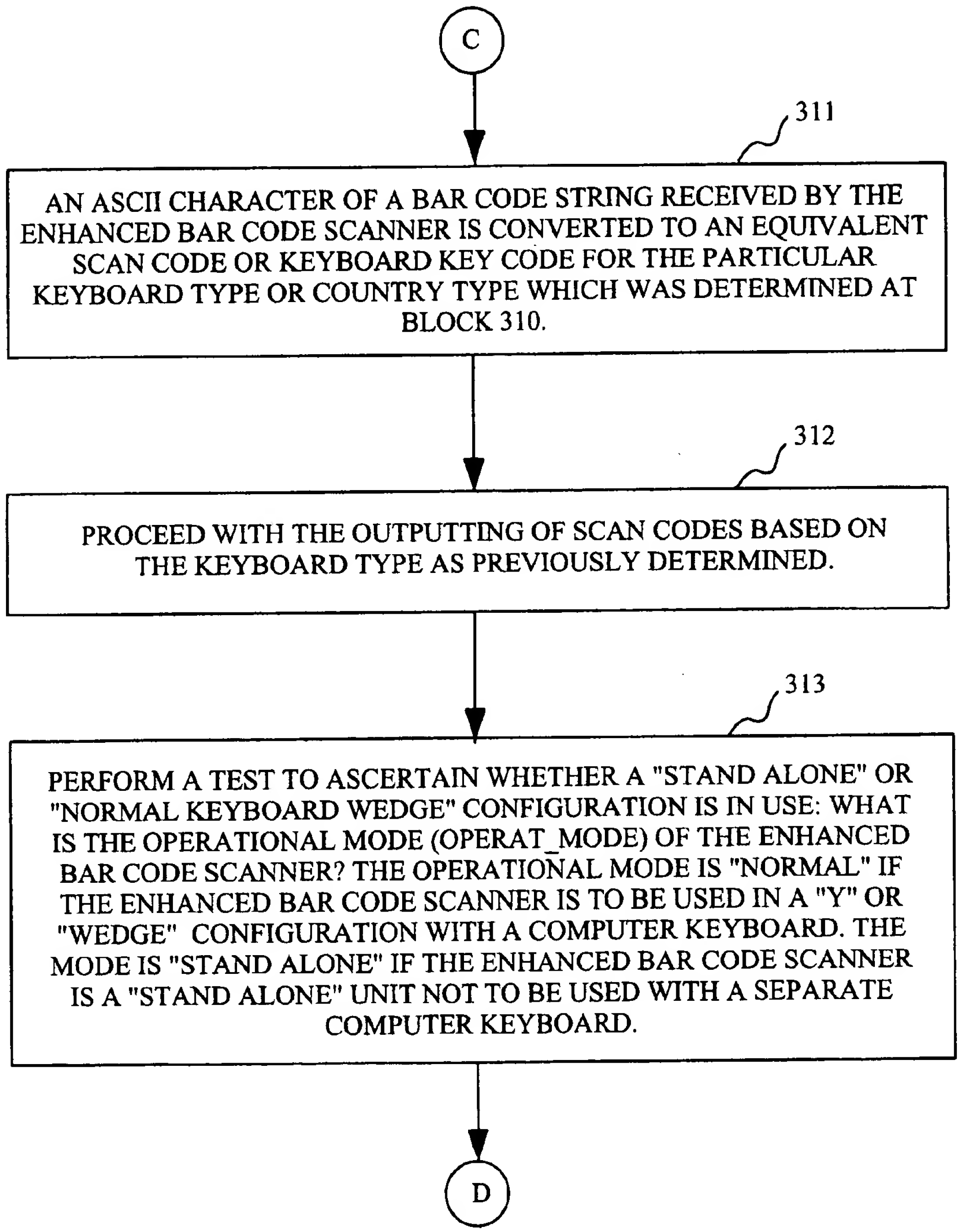

C
311
AN ASCII CHARACTER OF A BAR CODE STRING RECEIVED BY THE ENHANCED BAR CODE SCANNER IS CONVERTED TO AN EQUIVALENT SCAN CODE OR KEYBOARD KEY CODE FOR THE PARTICULAR KEYBOARD TYPE OR COUNTRY TYPE WHICH WAS DETERMINED AT BLOCK 310.
312
PROCEED WITH THE OUTPUTTING OF SCAN CODES BASED ON THE KEYBOARD TYPE AS PREVIOUSLY DETERMINED.
313
PERFORM A TEST TO ASCERTAIN WHETHER A "STAND ALONE" OR "NORMAL KEYBOARD WEDGE" CONFIGURATION IS IN USE: WHAT IS THE OPERATIONAL MODE (OPERAT_MODE) OF THE ENHANCED BAR CODE SCANNER? THE OPERATIONAL MODE IS "NORMAL" IF THE ENHANCED BAR CODE SCANNER IS TO BE USED IN A "Y" OR "WEDGE" CONFIGURATION WITH A COMPUTER KEYBOARD. THE MODE IS "STAND ALONE" IF THE ENHANCED BAR CODE SCANNER IS A "STAND ALONE" UNIT NOT TO BE USED WITH A SEPARATE COMPUTER KEYBOARD.
D

FIG. 3D

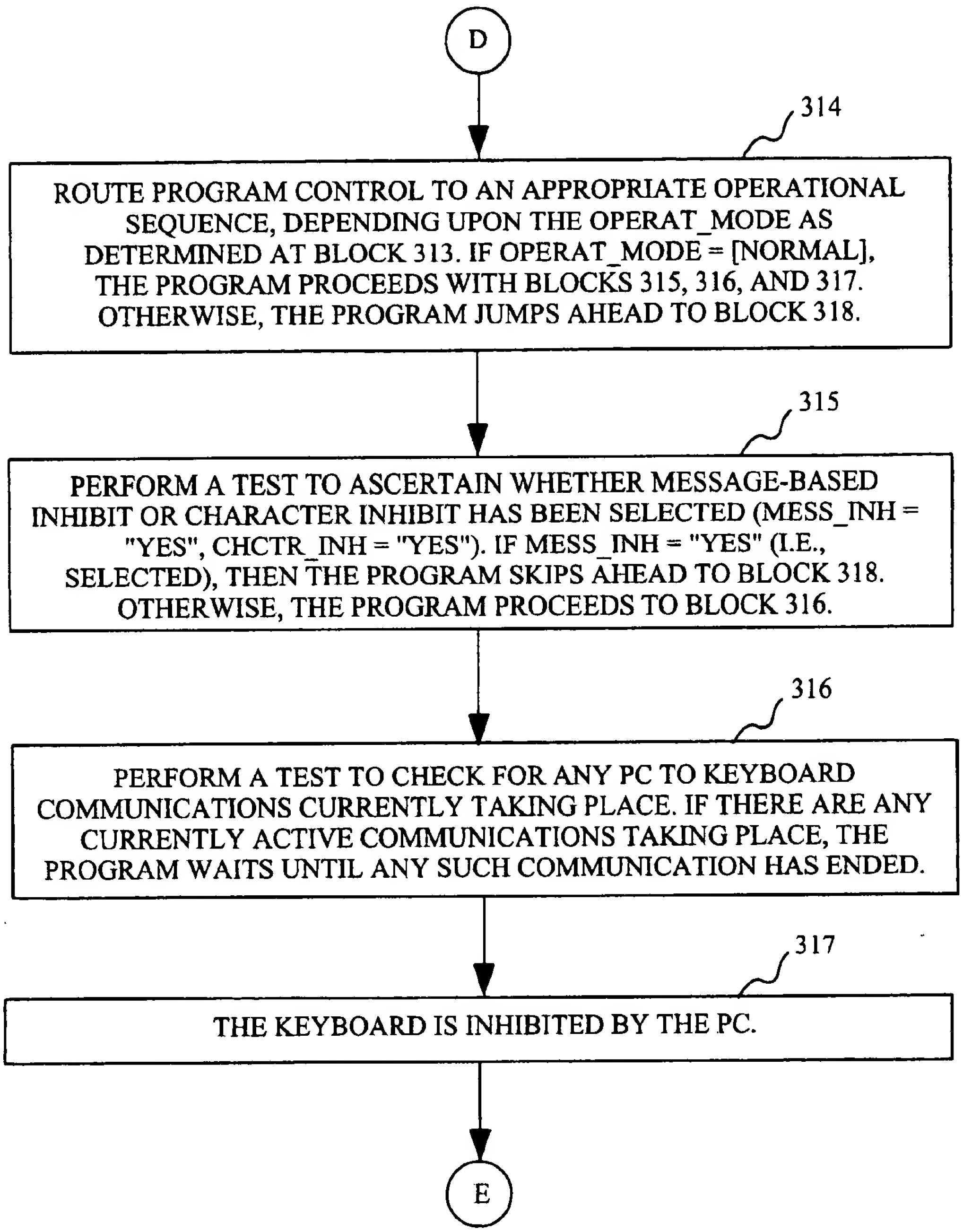

D
314
ROUTE PROGRAM CONTROL TO AN APPROPRIATE OPERATIONAL SEQUENCE, DEPENDING UPON THE OPERAT_MODE AS DETERMINED AT BLOCK 313. IF OPERAT_MODE = [NORMAL], THE PROGRAM PROCEEDS WITH BLOCKS 315, 316, AND 317. OTHERWISE, THE PROGRAM JUMPS AHEAD TO BLOCK 318.
315
PERFORM A TEST TO ASCERTAIN WHETHER MESSAGE-BASED INHIBIT OR CHARACTER INHIBIT HAS BEEN SELECTED (MESS_INH = "YES", CHCTR_INH = "YES"). IF MESS_INH = "YES" (I.E., SELECTED), THEN THE PROGRAM SKIPS AHEAD TO BLOCK 318. OTHERWISE, THE PROGRAM PROCEEDS TO BLOCK 316.
316
PERFORM A TEST TO CHECK FOR ANY PC TO KEYBOARD COMMUNICATIONS CURRENTLY TAKING PLACE. IF THERE ARE ANY CURRENTLY ACTIVE COMMUNICATIONS TAKING PLACE, THE PROGRAM WAITS UNTIL ANY SUCH COMMUNICATION HAS ENDED.
317
THE KEYBOARD IS INHIBITED BY THE PC.
E

FIG. 3E

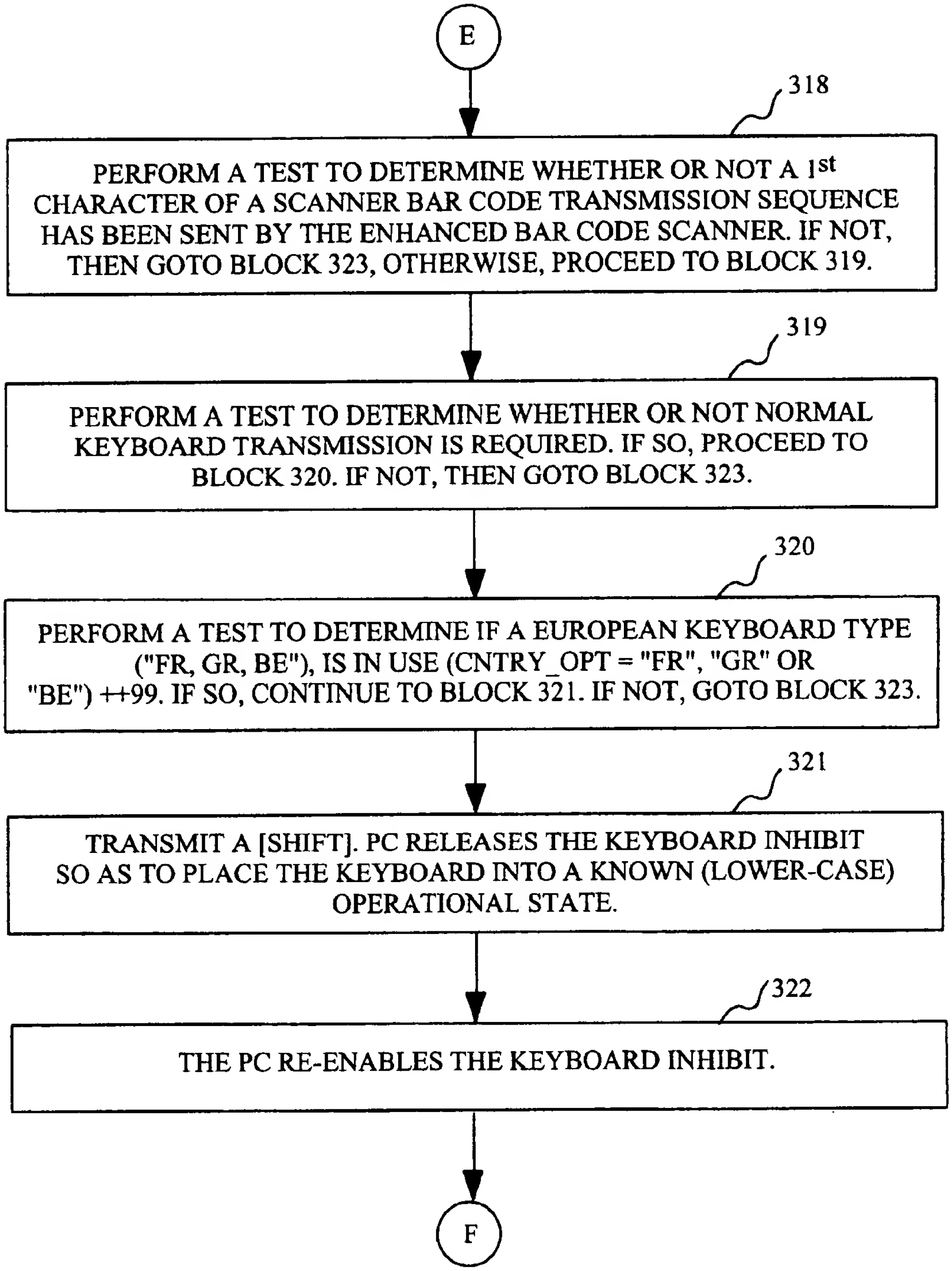

E
318
PERFORM A TEST TO DETERMINE WHETHER OR NOT A 1st CHARACTER OF A SCANNER BAR CODE TRANSMISSION SEQUENCE HAS BEEN SENT BY THE ENHANCED BAR CODE SCANNER. IF NOT, THEN GOTO BLOCK 323, OTHERWISE, PROCEED TO BLOCK 319.
319
PERFORM A TEST TO DETERMINE WHETHER OR NOT NORMAL KEYBOARD TRANSMISSION IS REQUIRED. IF SO, PROCEED TO BLOCK 320. IF NOT, THEN GOTO BLOCK 323.
320
PERFORM A TEST TO DETERMINE IF A EUROPEAN KEYBOARD TYPE ("FR, GR, BE"), IS IN USE (CNTRY_OPT = "FR", "GR" OR "BE") ++99. IF SO, CONTINUE TO BLOCK 321. IF NOT, GOTO BLOCK 323.
321
TRANSMIT A [SHIFT]. PC RELEASES THE KEYBOARD INHIBIT SO AS TO PLACE THE KEYBOARD INTO A KNOWN (LOWER-CASE) OPERATIONAL STATE.
322
THE PC RE-ENABLES THE KEYBOARD INHIBIT.
F

FIG. 3F

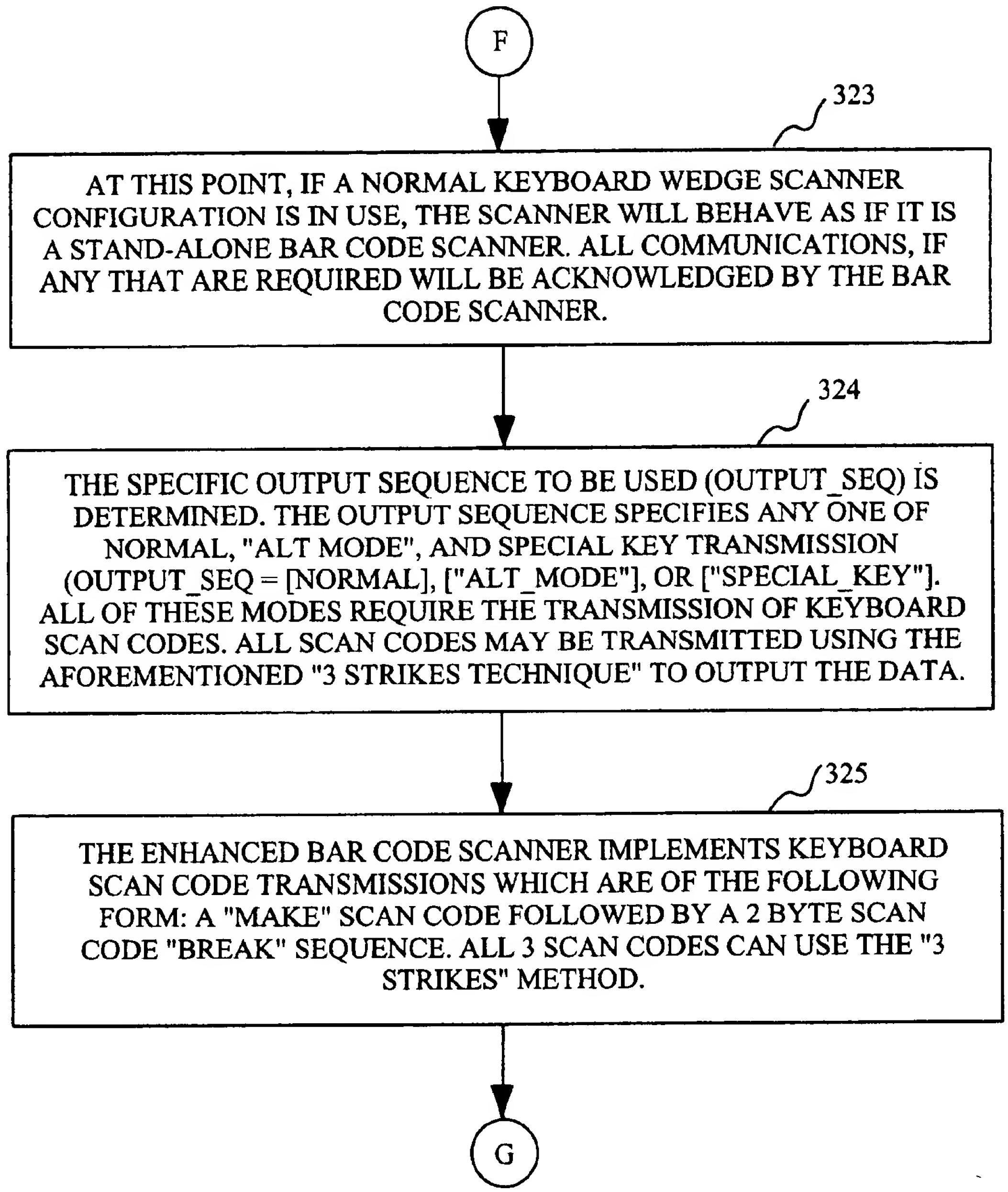

F
323
AT THIS POINT, IF A NORMAL KEYBOARD WEDGE SCANNER CONFIGURATION IS IN USE, THE SCANNER WILL BEHAVE AS IF IT IS A STAND-ALONE BAR CODE SCANNER. ALL COMMUNICATIONS, IF ANY THAT ARE REQUIRED WILL BE ACKNOWLEDGED BY THE BAR CODE SCANNER.
324
THE SPECIFIC OUTPUT SEQUENCE TO BE USED (OUTPUT_SEQ) IS DETERMINED. THE OUTPUT SEQUENCE SPECIFIES ANY ONE OF NORMAL, "ALT MODE", AND SPECIAL KEY TRANSMISSION (OUTPUT_SEQ = [NORMAL], ["ALT_MODE"], OR ["SPECIAL_KEY"]. ALL OF THESE MODES REQUIRE THE TRANSMISSION OF KEYBOARD SCAN CODES. ALL SCAN CODES MAY BE TRANSMITTED USING THE AFOREMENTIONED "3 STRIKES TECHNIQUE" TO OUTPUT THE DATA.
325
THE ENHANCED BAR CODE SCANNER IMPLEMENTS KEYBOARD SCAN CODE TRANSMISSIONS WHICH ARE OF THE FOLLOWING FORM: A "MAKE" SCAN CODE FOLLOWED BY A 2 BYTE SCAN CODE "BREAK" SEQUENCE. ALL 3 SCAN CODES CAN USE THE "3 STRIKES" METHOD.
G

FIG. 3G

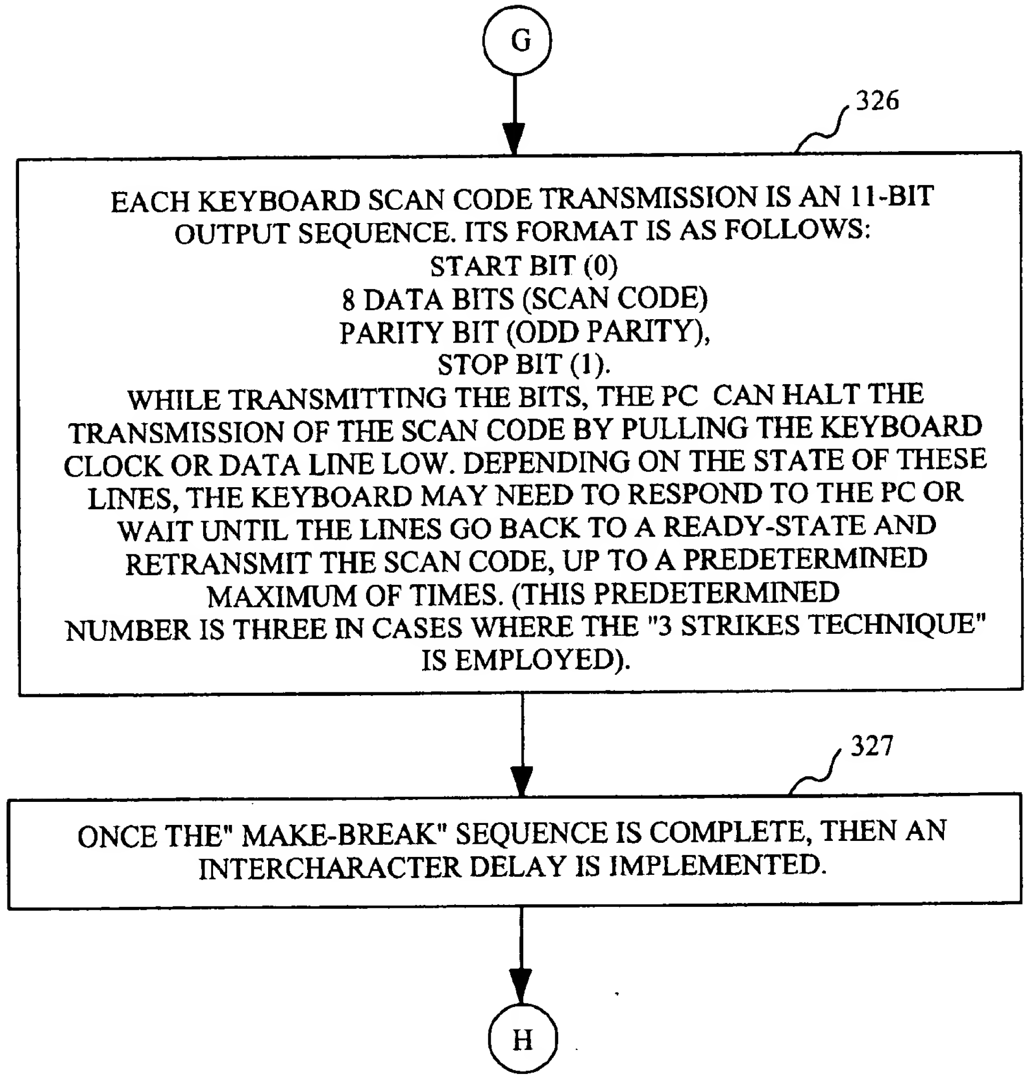


FIG. 3H

10/29/2003, EAST Version: 1.4.1

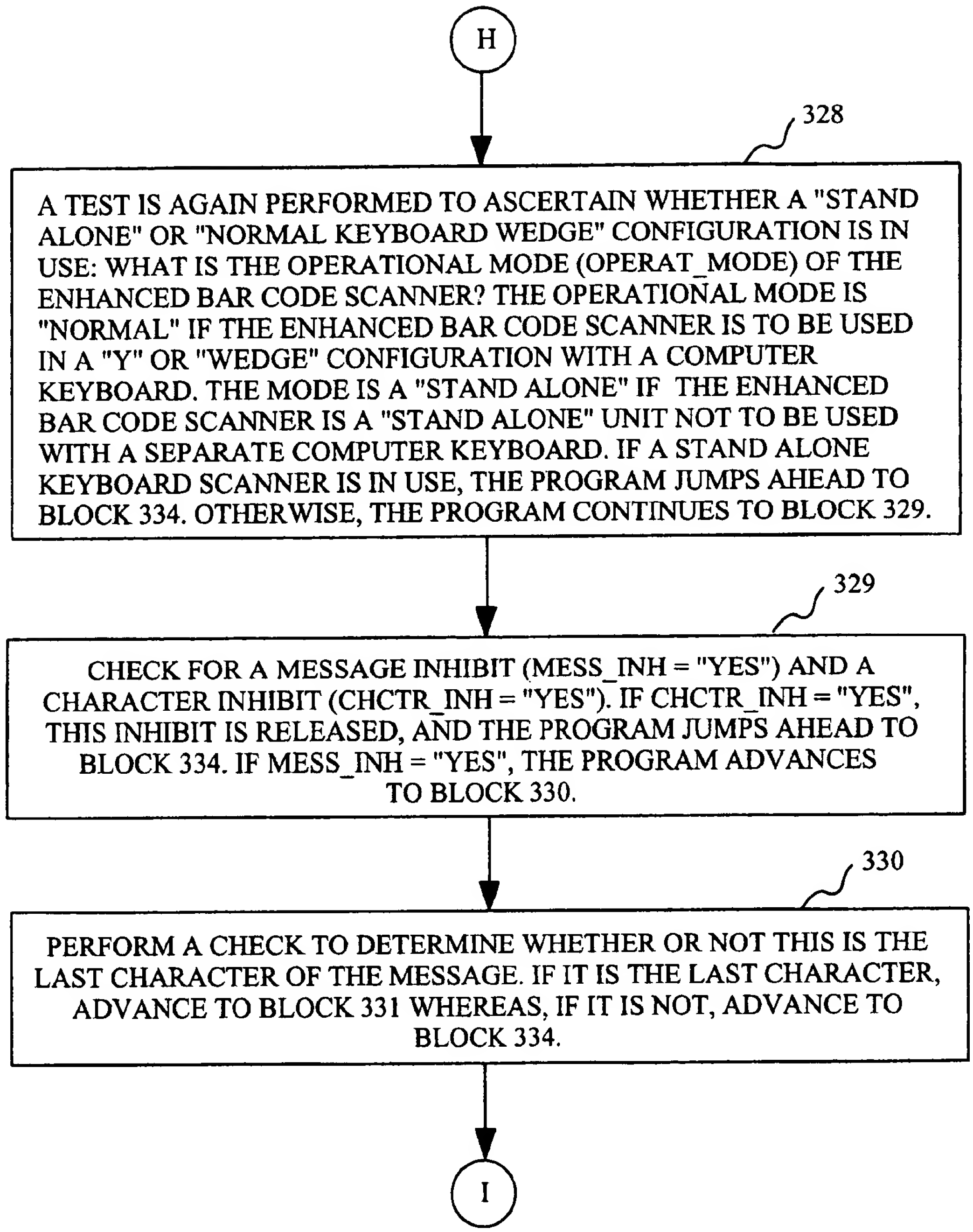


FIG. 3I

10/29/2003, EAST Version: 1.4.1

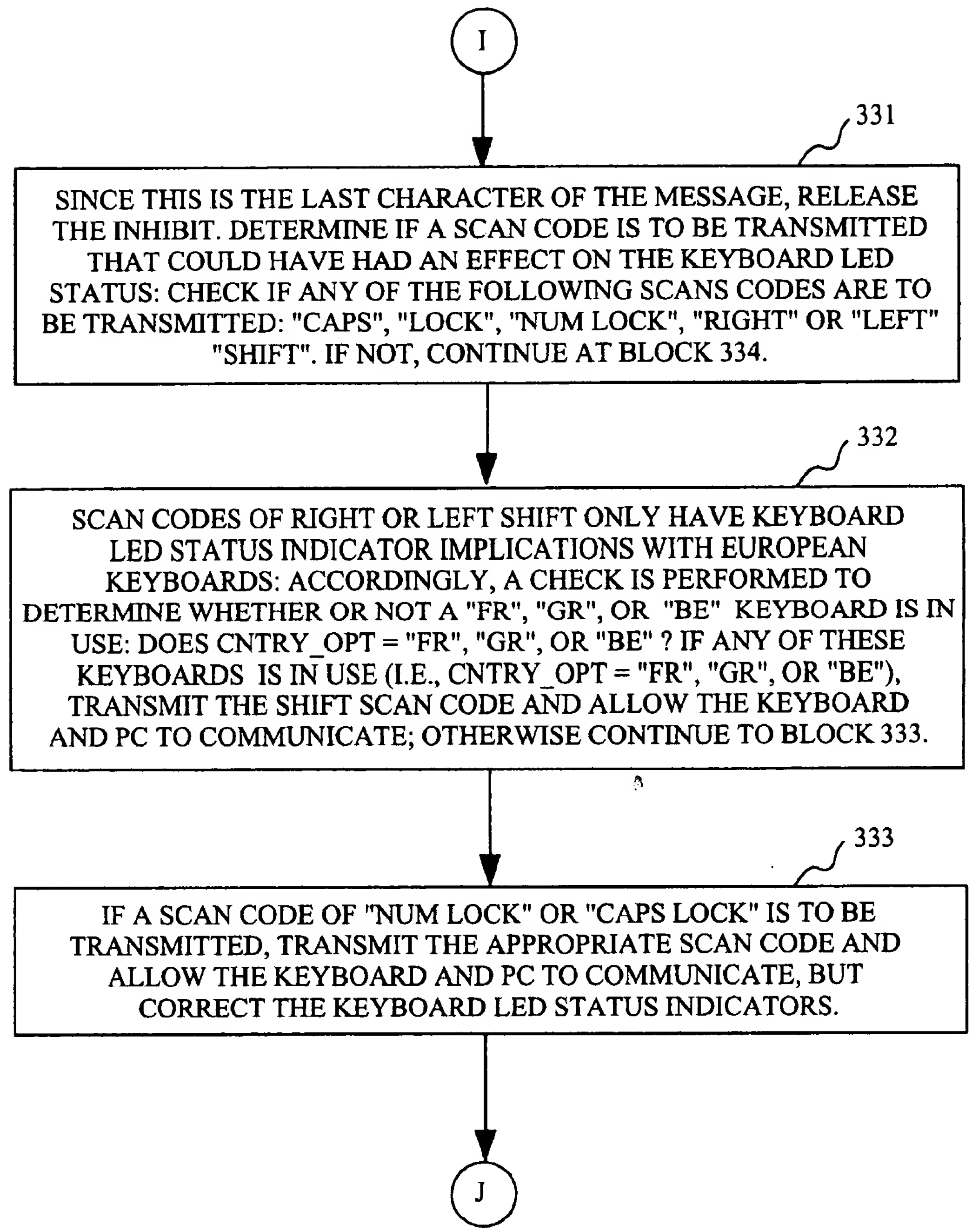

I
331
SINCE THIS IS THE LAST CHARACTER OF THE MESSAGE, RELEASE THE INHIBIT. DETERMINE IF A SCAN CODE IS TO BE TRANSMITTED THAT COULD HAVE HAD AN EFFECT ON THE KEYBOARD LED STATUS: CHECK IF ANY OF THE FOLLOWING SCANS CODES ARE TO BE TRANSMITTED: "CAPS", "LOCK", "NUM LOCK", "RIGHT" OR "LEFT" "SHIFT". IF NOT, CONTINUE AT BLOCK 334.
332
SCAN CODES OF RIGHT OR LEFT SHIFT ONLY HAVE KEYBOARD LED STATUS INDICATOR IMPLICATIONS WITH EUROPEAN KEYBOARDS: ACCORDINGLY, A CHECK IS PERFORMED TO DETERMINE WHETHER OR NOT A "FR", "GR", OR "BE" KEYBOARD IS IN USE: DOES CNTRY_OPT = "FR", "GR", OR "BE" ? IF ANY OF THESE KEYBOARDS IS IN USE (I.E., CNTRY_OPT = "FR", "GR", OR "BE"), TRANSMIT THE SHIFT SCAN CODE AND ALLOW THE KEYBOARD AND PC TO COMMUNICATE; OTHERWISE CONTINUE TO BLOCK 333.
333
IF A SCAN CODE OF "NUM LOCK" OR "CAPS LOCK" IS TO BE TRANSMITTED, TRANSMIT THE APPROPRIATE SCAN CODE AND ALLOW THE KEYBOARD AND PC TO COMMUNICATE, BUT CORRECT THE KEYBOARD LED STATUS INDICATORS.
J

FIG. 3J

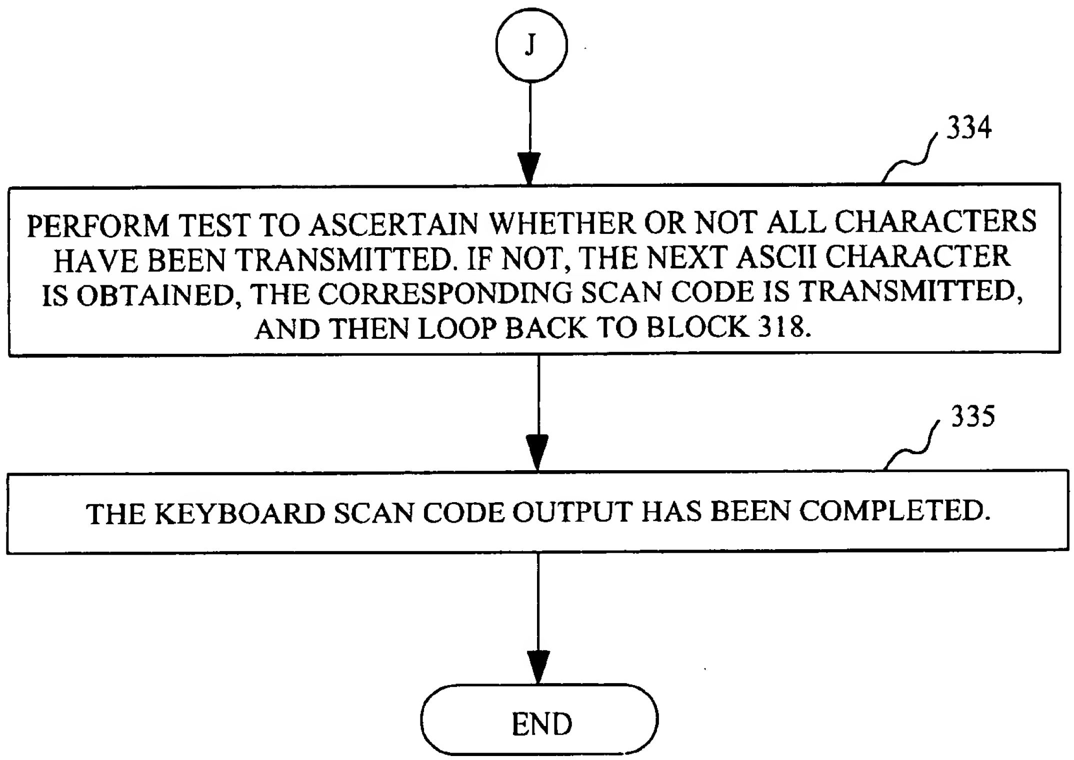

J
334
PERFORM TEST TO ASCERTAIN WHETHER OR NOT ALL CHARACTERS HAVE BEEN TRANSMITTED. IF NOT, THE NEXT ASCII CHARACTER IS OBTAINED, THE CORRESPONDING SCAN CODE IS TRANSMITTED, AND THEN LOOP BACK TO BLOCK 318.
335
THE KEYBOARD SCAN CODE OUTPUT HAS BEEN COMPLETED.
END

FIG 3K

TECHNIQUES FOR INTERFACING A BAR CODE SCANNER TO A PC USING A KEYBOARD RETRANSMIT PROTOCOL

The present patent application is based upon Provisional Application Ser. No. 60/144,361, filed on Jul. 16th, 1999, and entitled, "Techniques for Interfacing a Bar Code Scanner to a PC Using a Keyboard Retransmit Protocol", the disclosure of which is incorporated herein by reference.

FIELD OF THE INVENTION

The invention relates generally to bar code scanners, and more specifically, to techniques for interfacing bar code scanners with computers.

BACKGROUND OF THE INVENTION

For years, various bar code scanner manufacturers have been selling keyboard-wedge bar code scanners. With reference to FIG. **1**, bar code scanner **108** is connected to a personal computer (PC) **100** keyboard controller port **104** and a computer keyboard **110** in an Y or wedge type configuration. Bar code scanner **108** may contain an on-board processor **109**. PC **100** also contains a processor **101**. The Y or wedge type configuration is implemented using interconnection cable **106** such that computer keyboard **110** and bar code scanner **108** are placed in parallel across controller port **104**. This parallel configuration is used because keyboard controller **102** circuitry within presently-existing PCs (personal computers) **100** and laptop computers attempts to detect the existence of a computer keyboard **110** connected to the keyboard controller port **104**. The controller port **104** needs to be connected to a computer keyboard **110**, even if the keyboard is not to be used for subsequent data entry, and even if the controller port **104** is also connected to an input device other than a computer keyboard. If the keyboard controller **102** circuitry does not detect a keyboard connected to the controller port **104**, the PC **100** and/or laptop may then disable the port, preventing any further inputting of data. In the operational environment of FIG. **1**, this disablement poses a problem, because we desire to input further data as bar codes are detected and decoded.

With the increasing use of laptop computers and keyless data entry, the keyboard controller port shows great potential as a convenient, somewhat standardized, and readily available data input channel. However, this potential could be advantageously exploited only if it were possible to find some way around the necessity of connecting this port to an actual computer keyboard. By way of clarification, there are a number of existing programs that do not require the use of a computer keyboard per se, but these programs have neglected to provide mechanisms by which a computer keyboard is emulated, so as to prevent the controller port from being disabled.

Assume that a conventional keyboard wedge bar code scanner is connected to a keyboard controller port of a PC or laptop, as shown in FIG. **1**, while, at the same time, the keyboard that is connected in the parallel (Y) configuration is eliminated. Will the hardware configuration of FIG. **1** still function as desired? It is important to realize that keyboard-to-PC communications is implemented by means of a 2-way channel. Other types of data must be communicated between the PC and the keyboard in addition to information specifying the key or keys that were pressed. When a PC is powered up, the PC is programmed to check for the existence of a primary data input device, which is typically a keyboard. The PC begins a data exchange with the keyboard, and this communication is called "power-on diagnostics". If the keyboard is not present, or if the power-on diagnostics fail, the PC will not boot up. Accordingly, if a normal boot-up is desired, the keyboard shown in FIG. **1** should not be eliminated. In the case of laptop computers, a similar situation exists. A communication protocol is used to sense the presence of an external keyboard that is connected to the laptop's external keyboard port. If the laptop computer fails to detect a keyboard at the external keyboard port, then the laptop computer may disable its external keyboard port.

Even if a technique were to be developed by which a bar code scanner could successfully emulate a computer keyboard, another problem would then arise. Eleven (11)-bit transmission protocols are utilized almost universally to provide keyboard to PC data transfer. The transmission protocol begins with a Start bit (low), followed by 8 data bits, a Parity bit (odd parity) and finally a Stop bit (high). In the context of a computer keyboard, these 8 data bits are used to represent one or more scan codes corresponding to keyboard key presses. However, in the operational environment of FIG. **2**, these data bits can be employed to represent all or part of a detected bar code. Irrespective of whether these 8 bits represent actual key presses or bar codes, the aforementioned transmission protocol remains unchanged. The protocol allows the PC (or laptop) to interrupt the transmitted sequence up through the 9th bit. Since the 9th bit can be used to represent all or part of a scan code, the PC will prevent the communication of a scan code if the PC sends out a keyboard port inhibit any time before the 9th bit is received by the port. If the scan code represents all or part of a decoded bar code, upon issuance of a keyboard controller port interrupt, the PC will prevent the successful receipt of this bar code at the keyboard controller port.

The aforementioned keyboard inhibit problem is described in greater detail in a reference book entitled, "PC KEYBOARD DESIGN", by J. Konzak, and published by Annabook. In the configuration of FIG. **1**, the inhibit problem could cause decoded bar code data to be lost, ignored, corrupted, or misread once the PC or laptop interrupts communication. With the advent of multitasking operating systems, sophisticated network operating systems, and dual-keyboard port laptops and PC's, the problem worsens. Some of these operating systems interrupt keyboard port communications on a frequent and periodic basis, such as once every ten milliseconds. Of course, in operational environments where the keyboard controller port is not used with an auxiliary input devices, the computer keyboard will sense this stoppage of communications and retransmit the scan code after the PC releases its halt or inhibit of the keyboard. Existing bar code scanners are not so equipped. If a data transmission from a bar code scanner is interrupted, the scanner has no mechanism by which to ascertain whether or not a data entry error has occurred.

Pursuant to prior art techniques, bar code scanners had not monitored the inhibit signal because the decoded bar codes were wedged into the transmitted data for brief periods of time, relative to typed-in keyboard data. Also, as a practical matter, the keyboard BIOS virtually never inhibits the keyboard during scan code transmission. Any problems that may have been encountered were handled by changing certain programmable parameters such as inter-character delays or inter-scan-code delays.

SUMMARY OF THE INVENTION

A bar code scanner is equipped with a keyboard emulation mechanism so as to provide an enhanced interface to a

keyboard controller port of a computing device such as a personal or laptop computer. The keyboard emulation mechanism eliminates the necessity of connecting an actual computer keyboard to the port by responding to the computing device's standard power-on diagnostic procedure as if it were effectively an electrical equivalent of a computer keyboard.

Pursuant to a further embodiment of the invention, the keyboard emulation mechanism is equipped to detect a keyboard inhibit signal at the keyboard controller port. This inhibit signal may be generated by the computing device. If the keyboard emulation mechanism detects a keyboard inhibit signal while a data byte is being transmitted to the keyboard controller port, the keyboard emulation mechanism stops transmitting the data byte, waits for the inhibit signal to cease, and then retransmits the data byte to the keyboard controller port. This retransmission process is repeated up to a specified number of times, so as to provide additional opportunities for the data byte to be inputted to the keyboard controller port if the port is momentarily disabled by the keyboard inhibit signal. For many applications, it is advantageous to repeat the retransmission process up to three times for a given data byte. If the data byte is still not successfully received after the third attempt, the process is no longer repeated.

Pursuant to a still further embodiment of the invention, the keyboard emulation mechanism keeps track of status indications transmitted by the computing device and related to the status of one or more indicators on a computer keyboard. These indicators may be provided in the form of one or more LED lamps indicative of the status of the Caps-lock, Num-lock, and Scroll-lock keys. The keyboard emulation mechanism also keeps track of the transmission protocol currently in use, such as make-Break,Make-Only, AT or PS2-type scan-code transmission protocols. These status indications and transmission protocols may vary, depending upon the keyboard country type selected, and are maintained by the keyboard emulation mechanism so that the bar code scanner will transmit accurately to the computing device.

The keyboard emulation mechanism may be provided in the form of enhanced software, firmware, and/or programming instructions. This software, firmware, and/or operating instructions can be configured for use with presently-existing keyboard wedge scanners, as well as any of various scanners which may be developed in the future. One purpose of the software, firmware, and/or operating instructions is to accurately transmit a bar code on a transmission line that is subjected to possible halt or inhibit signals issued by the computing device It should be noted that any number of sources contained within the computing device could generate these inhibit signals. When a data byte is received at the keyboard port, this byte, which may represent all or part of a decoded bar code, is optionally displayed on a monitor coupled to the computing device.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a hardware block diagram showing a bar code scanner to personal computer interface using prior art techniques.

FIG. 2 is a hardware block diagram showing a bar code scanner to personal computer interface using the techniques of the present invention.

FIGS. 3A–3K together comprise a flowchart setting forth keyboard emulation techniques according to a preferred embodiment of the invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

FIG. 2 is a hardware block diagram showing a bar code scanner to personal computer interface using the techniques of the present invention. Enhanced bar code scanner **208** is connected to a keyboard controller port **204** of a personal computer **200** via interconnection cable **206**. Keyboard controller port **204** is driven by keyboard controller **202**. Interconnection cable **206** is shown for illustrative purposes only, as any other technique for conveying information from one place to another may be used in lieu of, or in addition to, interconnection cable **206**. These techniques may include wireless communication, wired communication, optical communication, and others. Moreover, PC **200** is shown for illustrative purposes, as any of a variety of computing devices may be employed in lieu of, or in addition to, PC **200**. In the example of FIG. 2, PC **200** includes a processor **205**. Enhanced bar code scanner **208** may contain an on-board processor **209**.

In the configuration of FIG. 2, a computer keyboard **110** need not be employed to perform the keyboard emulation techniques discussed herein. Accordingly, a wedge or Y connection between enhanced bar code scanner **208**, keyboard controller port **204**, and a computer keyboard is not required. However, if one wished to provide an additional input source to PC **200**, an optional computer keyboard could be provided. This optional keyboard could be connected, for example, using the Y or wedge type configuration shown in FIG. 1.

Enhanced bar code scanner **208** is "enhanced" in the sense that it is equipped with a keyboard emulation mechanism. This mechanism provides an interface to a keyboard port of a computing device such as a personal or laptop computer. As stated above, the keyboard emulation mechanism may include software, firmware, and/or operating instructions. Use of the term "keyboard emulation mechanism" is not necessarily indicative of any hardware enhancements to the bar code scanner. The keyboard emulation mechanism eliminates the necessity of connecting an actual computer keyboard to the port by responding to the computing device's standard power-on diagnostic procedure as if it were effectively an electrical equivalent of a computer keyboard.

Pursuant to a further embodiment of the invention, the keyboard emulation mechanism of enhanced bar code scanner **208** is equipped to detect a keyboard inhibit signal at the keyboard controller port **204**. This inhibit signal may be generated by PC **200**), and/or by any of a number of devices coupled to, and/or within this computing device. For example, the inhibit signal may be generated by the PC pulling the clock line to a logic "low" state. In this example, the keyboard emulation mechanism would detect the existence of a keyboard inhibit signal by monitoring the clock line.

If the keyboard emulation mechanism detects a keyboard inhibit signal while a data byte is being transmitted to the keyboard controller port **204**, the keyboard emulation mechanism retransmits this data byte to the keyboard controller port **204**. This retransmission process is repeated up to a specified number of times, so as to provide additional opportunities for the data byte to be inputted to the keyboard controller port **204** if the port is momentarily disabled by the keyboard inhibit signal. For many applications, it is advantageous to repeat the retransmission process up to three times for a given data byte. If the data byte is still not successfully received after the third attempt, the process is

10/29/2003, EAST Version: 1.4.1

no longer repeated. This data byte may represent one or more scan codes which, in turn, represent one or more keyboard key presses or decoded bar codes.

Pursuant to a still further embodiment of the invention, the keyboard emulation mechanism keeps track of status indications transmitted by the computing device (in the example of FIG. 2, PC 200) and related to the status of one or more indicators on a computer keyboard. These indicators may be provided in the form of one or more LED lamps indicative of the status of the Caps-lock, Num-lock, and Scroll-lock keys. The keyboard emulation mechanism also keeps track of the transmission protocol currently in use, such as make-Break,Make-Only, AT or PS2-type scan-code transmission protocols. These status indications and transmission protocols may vary, depending upon the keyboard country type selected, and are maintained by the keyboard emulation mechanism so that the enhanced bar code scanner 208 will transmit accurately to PC 200.

The keyboard emulation mechanism may be provided in the form of enhanced software, firmware, and/or programming instructions. This software, firmware, and/or operating instructions can be configured for use with presently-existing keyboard wedge scanners, as well as any of various scanners which may be developed in the future. One purpose of the software, firmware, and/or operating instructions is to accurately transmit a bar code on a transmission line that is subjected to possible halt or inhibit signals issued by the computing device. It should be noted that any number of sources contained within the computing device could generate these inhibit signals. When a data byte is received at the keyboard port, this byte, which may represent all or part of a decoded bar code, is optionally displayed on a monitor coupled to the computing device.

Refer to FIGS. 3A–3K which together comprise a flowchart setting forth a keyboard emulation method according to a preferred embodiment of the invention. In overview, the method is organized into a first sequence of steps related to power-up and diagnostics (blocks 301–304), a second sequence of steps related to the decoding and storing of data (blocks 305–306), a third sequence of steps related to keyboard output (blocks 307–308), a fourth sequence of steps related to ASCII to scan code conversion (blocks 309–311), a fifth sequence of steps related to preparation for a keyboard output sequence (blocks 312–317), a sixth sequence of steps related to first character keyboard initialization (blocks 318–322), a seventh sequence of steps related to keyboard scan code output (blocks 323–327), an eighth sequence of steps related to preparation for exiting the keyboard scan code transmission program (blocks 328–333), and a ninth sequence of steps related to a determination of whether or not there are more characters to transmit (blocks 334–335). It should be noted that the above-described organization of the steps into nine sequences is set forth for purposes of illustration and convenience. The steps may, but need not, be organized in the manner described above.

First Sequence of Steps

Power-Up and Power on Diagnostics

The program of FIGS. 3A–3K commences at block 301, where power is applied to enhanced bar code scanner (208, FIG. 2) and/or PC (200, FIG. 2) Next, at block 302 (FIG. 3), the program determines the operational mode of the enhanced bar code scanner. The operational mode is "normal" if the enhanced bar code scanner is to be used in a Y or wedge configuration with a computer keyboard, similar to the configuration of FIG. 1. The mode is "stand-alone" if, as shown in FIG. 2, the enhanced bar code scanner is a "stand alone" unit not to be used with a separate computer keyboard. If the operational mode is "normal", the program skips ahead to block 305. Otherwise, the program proceeds to block 303.

At block 303, it has already been determined that a separate computer keyboard (refer to FIG. 1) is not present. Accordingly, the enhanced bar code scanner (FIG. 2) must communicate to PC 200 all command bytes for successful completion of power-on diagnostics. The enhanced bar code scanner begins this keyboard emulation process by transmitting an "AA" to indicate that the "keyboard" self-test was completed successfully. After this point, the PC and "keyboard" (as emulated by the enhanced bar code scanner) may communicate and acknowledge any of a number of various commands to complete the power-on diagnostic sequence. Some PC's require more information than others for the power-on diagnostic procedure. Therefore the number of commands sent by the PC and acknowledged by the emulated keyboard is variable. These commands may include any of the following: Typamatic Rate, Make/Break status, scan code set enabled, and LED status.

At block 304, once the sequence of command bytes of block 303 are completed, the PC will power up normally. Note that powering up is sometimes referred to as "booting up". Without successful completion of these commands, the PC will fail to power up. After the PC boots up, the enhanced bar code scanner is ready to decode data.

Second Sequence of Steps

Decoding and Storing Data

At block 305, bar code symbols are decoded into human-readable and/or ASCII characters which are then stored in a memory device associated with the enhanced bar code scanner, such as, for example, random-access memory (RAM). At block 306, the program generates an appropriate output interface equipped to select output data from the enhanced bar code scanner.

Third Sequence of Steps

Begin Keyboard Output

A test is performed at block 307 to determine the output mode of the emulated keyboard. Typical output modes include AT, XT, and PS2. At block 308, the program determines the type of keyboard transmission that is to take place. Types of keyboard transmission include normal, ALT Mode, and special key transmission sequences.

Fourth Sequence of Steps

ASCII to Scan Code Conversion

At block 309, if the type of keyboard transmission (as determined at block 308) is normal transmission, the program proceeds to blocks 310 & 311. Otherwise, the program jumps ahead to block 312. The country output setting of the keyboard is determined at block 310. This country output setting specifies any of a plurality of keyboards, such as US (United States), UK (United Kingdom), FR (France), GR (Germany), BE (Belgium), SP (Spain), IT (Italy), and JP (Japan). Next, at block 311, an ASCII character of a bar code string received by the enhanced bar code scanner is converted to an equivalent scan code or keyboard key code for the particular keyboard type or country type determined at block 310.

Fifth Sequence of Steps

Prepare for Keyboard Output Sequence

The program proceeds (block 312) with the outputting of scan codes based on the keyboard type as previously deter-

mined at block 310. Then, at block 313, a test is performed to ascertain whether a "stand alone" (FIG. 2) or "normal keyboard wedge" configuration is in use. The "normal keyboard wedge" configuration is similar to FIG. 1, except that an enhanced bar code scanner is used in place of bar code scanner 108. At block 314, program control is routed to an appropriate operational sequence, depending upon the configuration in use as determined at block 313. If a "normal keyboard wedge" configuration is in use, the program proceeds with blocks 315, 316, and 317. Otherwise, the program jumps ahead to block 318.

At block 315, a test is performed to ascertain whether message-based inhibit or character inhibit has been selected. If message inhibit has been selected, then the program skips ahead to block 318. Otherwise, the program proceeds to block 316. A test is performed at block 316 to check for any PC to keyboard communications currently taking place. If there are any currently active communications taking place, the program waits until any such communication has ended. Then (block 317), the keyboard is inhibited from the PC.

Sixth Sequence of Steps

First Character Keyboard Initialization

This sequence commences by performing a test to determine whether or not a 1ST character of a scanner bar code transmission sequence has been sent by the enhanced bar code scanner (block 318). This scanner bar code transmission sequence is organized into a data byte, as previously described. If not, the program jumps ahead to block 323. Otherwise, the program proceeds to block 319, where a check is performed to determine whether or not normal keyboard transmission is required. If so, the program proceeds to block 320. If not, the program jumps ahead to block 323.

Next, a test is performed to determine if a European keyboard type (FR, GR, BE).is in use. If so, the program continues to block 321. If not, the program jumps ahead to block 323. At block 321, a [SHIFT] is transmitted and the PC releases the keyboard inhibit so as to place the keyboard into a known (lower-case) operational state. The PC then re-enables the keyboard inhibit (block 322).

Seventh Sequence of Steps

Keyboard Scan Code (S) Are Outputted

At this point (block 323), if a normal keyboard wedge scanner configuration is in use, the scanner will communicate with other system components as if it was a standalone bar code scanner. All communications, if any, that are required will be acknowledged by the bar code scanner. Next (block 324), the output sequence is determined. The output sequence specifies any of Normal, Alt Mode, and Special Key transmission. All of these modes require the transmission of keyboard scan codes. Alt mode transmits the decimal value of the ASCII value using the numeric keypad scan codes. Special Key transmissions are the transmission of non-ASCII keys (i.e. function keys—F1, F2, etc., arrow keys and so forth). Normal key transmission is the transmission of normal ASCII characters that have an associated keyboard scan code. All scan codes may be transmitted using the aforementioned "3 strikes technique" to output the data.

The enhanced bar code scanner implements a keyboard scan code transmission (block 325) using a plurality of data bytes. A "MAKE" scan code followed by a 2 byte scan code BREAK sequence is transmitted. All 3 scan codes can use the "3 strikes" method. At block 326, each keyboard scan code transmission is an 11-bit output sequence. Its format is as follows: Start bit (0), 8 data bits (scan code), parity bit (odd parity) and a stop bit (1). While transmitting the bits, the PC can halt the transmission of the scan code by pulling the keyboard clock or data line low. Depending on the state of these lines, the keyboard may need to respond to the PC or wait until the lines go back to a ready-state and retransmit th e scan c ode , up to a predetermined maximum number of times. This predetermined number is three in cases where the "3 strikes technique" is employed. Once the MAKE-BREAK sequence is complete (block 327), then an inter-character delay is implemented.

Eighth Sequence of Steps

Prepare to Exit Keyboard Scan Code Transmission Routine

A test is again performed to ascertain whether a "stand alone" (FIG. 2) or "normal keyboard wedge" configuration is in use (block 328). If a Stand alone keyboard scanner is in ue, the program jumps ahead to block 334. Otherwise, the program continues to block 329, where the program checks for a Message Inhibit and a Character Inhibit. If a Character Inhibit exists, this inhibit is released, and the program jumps ahead to block 334. If a Message Inhibit exists, the program advances to block 330, where a check is performed to determine whether or not this is the last character of the message. If it is the last character , the program advances to block 331 whereas, if it is not., the program advances to block 334.

At block 331, since this is the last character of the message, it is time to release the inhibit. It must also be determined if a scan code is to be transmitted that could have had an effect on the keyboard LED status. Check if any of the following scans codes are to be transmitted: Caps: Lock, Num Lock, Right or Left Shift. If not, continue at Block 334. Next (block 332), scan codes of Right or Left Shift only have keyboard LED status indicator implications with European keyboards. Accordingly, a check is performed to determine whether or not a FR, GR, or BE keyboard is in use. If any of these keyboards is in use, transmit the Shift scan code and allow the keyboard and PC to communicate; otherwise continue to block 333. If a scan code of Num Lock or Caps Lock is to be transmitted, transmit the appropriate scan code and allow the KB and PC to communicate, but correct the keyboard LED status indicators.

Ninth Sequence of Steps

Determine Last Character to be Transmitted

Check if all characters have been transmitted (block 334). If not, the next ASCII character is obtained, the corresponding scan code is transmitted, and the program loops back to block 318. At block 335, the keyboard scan code output has been completed.

The program of FIGS. 3A–3K has been organized into nine operational sequences for illustrative purposes and as an explanatory aid. Those skilled in the art can appreciate that the program need not necessarily be organized into nine operational sequences, as variations thereto are still within the spirit and scope of the invention.

As stated above, the keyboard emulation mechanism may include software, firmware, and/or operating instructions. Use of this term herein is not necessarily indicative of any hardware enhancements to the bar code scanner. If the scanner is programmed to allow for up to three retransmissions, such a method is referred to as the "3-strikes-technique". A conventional transmission of one pressed key consists of three (3) characters—a make scan code and two (2) break code characters. Therefore, this technique could possibly transmit up to 9 characters to complete the key sequence. This retransmit technique gives the enhanced bar code scanner a limited number of retries

per scan code, thus allowing a transmission sequence to move forward instead of hanging or sticking while trying to retransmit the same character over and over again.

But why should there ever be a need to retransmit a scan code? The retransmit protocol was developed after noticing that other devices or software within the PC could also inhibit the keyboard controller port. Some devices/software such as network drivers may inhibit the keyboard controller port every few milliseconds to perform some function without interrupt. Other devices, such as a mouse, may be on the same interrupt as the keyboard controller port, thus requiring the keyboard controller port to be inhibited while the software governing the mouse moves the mouse cursor. With laptops, the same types of inhibit issues can exist. However, even more restraints may be placed on the external keyboard controller port of a laptop than would be the case with a conventional PC. This renders use of the keyboard controller port as an auxiliary input channel all the more difficult in the operational environment of laptop computing.

With the advent of Windows 3.1 and Window 95, it appeared that more and more devices/software were effecting the behavior of the keyboard controller port. Therefore, it was imperative that a technique was developed that would transmit the scan code accurately, but take into account the fact that during transmission the keyboard could receive an inhibit and a retransmit would be necessary to accurately transmit the decoded bar code. The "3-strikes-technique" set forth herein accomplishes this task uniquely within a limited transmission sequence.

We claim:

1. A bar code scanner adapted for connection to: (i) a keyboard controller port of a computing device, and (ii) a computer keyboard, in a keyboard-wedge configuration, the bar code scanner having a keyboard inhibit mechanism for preventing an intermingling of bar code data acquired by the bar code scanner and typed-in keyboard data and typed into the keyboard; the keyboard inhibit mechanism comprising:

(a) a monitoring mechanism adapted to test for the existence of any communication in progress between the keyboard and the computing device;

(b) a switching mechanism, coupled to the monitoring mechanism, wherein, in response to an absence of communication in progress between the keyboard and the computing device, the switching enters a first state so as to disable subsequent communications between the computer keyboard and the keyboard controller port, thereby inhibiting the keyboard;

(c) a bar code transmission mechanism, responsive to the switching mechanism being in the first state, for transmitting a decoded bar code to the keyboard controller port as a sequence of data bytes, for emulating any communications that may be required between the keyboard controller port and the keyboard; and, upon an initial power-on of the computing device, providing an interface to a keyboard controller port of the computing device, so as to eliminate the necessity of connecting an actual computer keyboard to the keyboard controller port by responding to a power-on diagnostic procedure implemented by the computing device as if the bar code transmission mechanism were effectively an electrical equivalent of a computer keyboard; and wherein the switching mechanism is responsive to the bar code transmission mechanism such that, once the bar code transmission mechanism has transmitted all characters of the bar code to the keyboard controller port, the switching mechanism enters a second state so as to release the keyboard inhibit and to permit communications between the keyboard and the keyboard controller port.

2. The bar code scanner of claim **1** wherein the computing device is a personal computer (PC).

3. The bar code scanner of claim **1** wherein the computing device is a laptop computer.

4. The bar code scanner of claim **1** wherein the keyboard emulation mechanism is equipped to detect a keyboard inhibit signal at the keyboard controller port.

5. The bar code scanner of claim **4** wherein the keyboard inhibit signal is generated by the computing device.

6. The bar code scanner of claim **1** wherein, if the keyboard emulation mechanism detects a keyboard inhibit signal while a data byte including one or more scan codes is being transmitted to the keyboard controller port, the keyboard emulation mechanism retransmits the data byte to the keyboard controller port.

7. The bar code scanner of claim **6** wherein the keyboard emulation mechanism waits until the keyboard inhibit signal ceases and then retransmits the data byte to the keyboard controller port.

8. The bar code scanner of claim **6** wherein the retransmission of the data byte is repeated up to a specified number of times, so as to provide additional opportunities for the data byte to be inputted to the keyboard controller port if the port is momentarily disabled by the keyboard inhibit signal.

9. The bar code scanner of claim **8** wherein the retransmission of the data byte is repeated for up to three times for a given data byte and, if the data byte is still not successfully received after the third attempt, the retransmission is no longer repeated.

10. The bar code scanner of claim **6** wherein the data byte represents a scan code indicative of all or a portion of a decoded bar code.

11. The bar code scanner of claim **1** wherein the keyboard emulation mechanism further includes a tracking mechanism for keeping track of status indications transmitted by the computing device and related to the status of one or more visual indicators on a computer keyboard.

12. The bar code scanner of claim **11** wherein the visual indicators are provided in the form of one or more LED lamps indicative of the status of at least one of a Caps-lock key, a Num-lock key, and a Scroll-lock key.

13. The bar code scanner of claim **1** wherein the keyboard emulation mechanism further includes a protocol tracking mechanism for keeping track of a scan code transmission protocol currently in use.

14. The bar code scanner of claim **13** wherein the scan code transmission protocol includes any of Make-Break, Make-Only, AT and PS2 scan-code transmission protocols.

15. The bar code scanner of claim **1** wherein the keyboard emulation mechanism is provided in the form of at least one of software, firmware, and/or programming instructions.

16. The bar code scanner of claim **15** wherein the software, firmware, and/or operating instructions is configured for use with presently-existing keyboard wedge bar code scanners.

17. The bar code scanner of claim **15** wherein the software, firmware, and/or operating instructions are adapted to transmit a scan code to a keyboard controller port that is subjected to possible keyboard inhibit signals issued by the computing device, such that the scan code is received at the keyboard controller port subsequent to issuance of the keyboard inhibit signal.

18. The bar code scanner of claim **17** wherein the scan code is received at the keyboard controller port and then displayed on a monitor coupled to the computing device.

19. A method for use with a bar code scanner that is adapted for connection to: (i) a keyboard controller port of a computing dévice, and (ii) a computer keyboard, in a keyboard-wedge configuration, the method preventing an intermingling of bar code data acquired by the bar code scanner and typed-in keyboard data typed into the keyboard, the method performing the steps of:

(a) coupling the bar code scanner to the keyboard controller port of the computing device,

(b) the bar code scanner responding to a power-on diagnostic procedure implemented by the computing device as if the bar code scanner were effectively an electrical equivalent of a computer keyboard;

(c) the bar code scanner testing for the existence of any communication in progress between the keyboard and the computing device;

(d) if no such communication is in progress, the bar code scanner causing a switching mechanism to enter a first state so as to disable communications between the keyboard and the computer controller port, thereby inhibiting the keyboard;

(e) the bar code scanner transmitting a decoded bar code to the keyboard controller port as a sequence of data bytes, and/or emulating any communications that may be required between the keyboard controller port and the keyboard; and

(f) once the bar code scanner has transmitted all characters of the bar code to the keyboard controller port, the bar code scanner releasing the keyboard inhibit by causing the switching mechanism to enter a second state, so as to permit communications between the keyboard and the keyboard controller port.

20. The method of claim 19 wherein the computing device is a personal computer (PC).

21. The method of claim 19 wherein the computing device is a laptop computer.

22. The method of claim 19 further including the step of the bar code scanner detecting a keyboard inhibit signal at the keyboard controller port.

23. The method of claim 22 wherein the keyboard inhibit signal is generated by the computing device.

24. The method of claim 22 further including the steps of:

(a) the bar code scanner detecting a keyboard inhibit signal while a data byte specifying a scan code is being transmitted to the keyboard controller port, and

(b) the bar code scanner retransmitting this data byte to the keyboard controller port.

25. The method of claim 24 further including the steps of:

(a) the keyboard emulation mechanism waiting until the keyboard inhibit signal ceases, and then

(b) retransmitting the data byte to the keyboard controller port.

26. The method of claim 24 wherein step (b) is repeated up to a specified number of times, so as to provide additional opportunities for the scan code to be inputted to the keyboard controller port if the port is momentarily disabled by the keyboard inhibit signal.

27. The method of claim 26 wherein the retransmission of the data byte is repeated for up to three times for a given data byte and, if the data byte is still not successfully received after the third attempt, the retransmission is no longer repeated.

28. The method of claim 24 wherein the scan code represents a decoded bar code.

29. The method of claim 19 further including the step of the bar code scanner keeping track of status indications transmitted by the computing device and related to the status of one or more visual indicators on a computer keyboard.

30. The method of claim 29 wherein the visual indicators are provided in the form of one or more LED lamps indicative of the status of at least one of a Caps-lock key, a Num-lock key, and a Scroll-lock key.

31. The method of claim 19 further including the step of the bar code scanner keeping track of a scan code transmission protocol currently in use.

32. The method of claim 30 wherein the scan code transmission protocol includes any of Make-Break,Make-Only, AT and PS2 scan-code transmission protocols.

33. The method of claim 19 implemented by of at least one of software, firmware, and/or programming instructions.

34. The method of claim 23 wherein the software, firmware, and/or operating instructions is configured for use with presently-existing keyboard wedge scanners.

35. The method of claim 33 wherein the software, firmware, and/or operating instructions are adapted to transmit a scan code to a keyboard controller port that is subjected to possible keyboard inhibit signals issued by the computing device, such that the scan code is received at the keyboard controller port subsequent to issuance of the keyboard inhibit signal.

36. The method of claim 35 wherein the scan code is received at the keyboard controller port and then displayed on a monitor coupled to the computing device.

* * * * *

US006536666B1

(12) United States Patent
Hudrick

(10) Patent No.: US 6,536,666 B1
(45) Date of Patent: Mar. 25, 2003

(54) **TECHNIQUES FOR INTERFACING A BAR CODE SCANNER TO A PC USING A MESSAGE-BASED AND/OR CHARACTER-BASED KEYBOARD INHIBIT**

(75) Inventor: **Donald T. Hudrick**, Sicklerville, NJ (US)

(73) Assignee: **Metrologic Instruments, Inc.**, Blackwood, NJ (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/616,776**

(22) Filed: **Jul. 14, 2000**

Related U.S. Application Data

(60) Provisional application No. 60/144,389, filed on Jul. 16, 1999.

(51) **Int. Cl.**^{7} **G06K 7/10**
(52) **U.S. Cl.** **235/462.15**
(58) **Field of Search** 235/462.15

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,021,640 A	*	6/1991	Muroi	186/61
5,178,234 A		1/1993	Sakurai et al.	
5,848,292 A	*	12/1998	Nathan	710/2
5,854,945 A	*	12/1998	Criscito et al.	235/435
5,875,415 A		2/1999	Lieb et al.	
6,263,383 B1	*	7/2001	Lee et al.	709/218
6,312,175 B1	*	11/2001	Lum	400/472

FOREIGN PATENT DOCUMENTS

JP 8-328717 * 12/1996

OTHER PUBLICATIONS

Copyright 1996 Information Access Company; Automatic I.D. News; Sep., 1996; p. 18; ISSN: 0890–9768.

Copyright 1996 Information Access Company; Packaging Digest; May, 1996; vol. 33; No. 6; p. 34; ISSN: 0030–9117.

MS900 by Metrologic Instruments, Inc.; Sep. 1998; www-.metrologic.com.

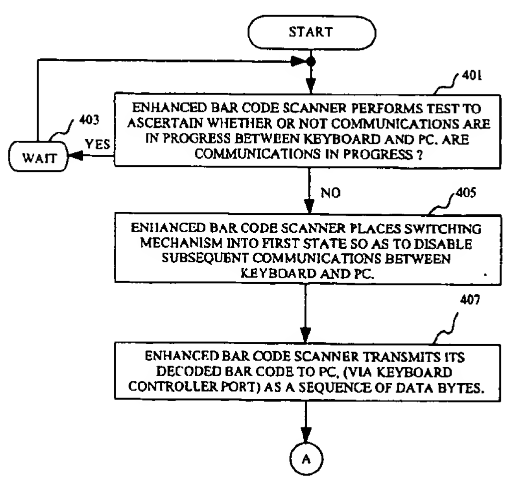
* cited by examiner

Primary Examiner—Mark Tremblay
(74) *Attorney, Agent, or Firm*—Morgan, Lewis & Bockius, LLP

(57) **ABSTRACT**

The intermingling of bar code data and typed-in data in a keyboard-wedge configuration is prevented through the use of a message-based keyboard inhibit procedure implemented by the bar code scanner. In this manner, both the typed-in data and the bar code data will remain uncorrupted, even if the keyboard data is being entered substantially simultaneously with the scanning and/or decoding of bar code data. This message-based keyboard inhibit procedure tests for any communication in progress between the keyboard and PC. If no communication is in progress, the bar code scanner places a switching mechanism into a first state so as to disable communications between the keyboard and a keyboard controller port of a PC (personal computer), thereby inhibiting the keyboard. The bar code scanner then transmits the decoded bar code to the PC as a sequence of data bytes. The bar code scanner also implements any communications that are required between the PC and the keyboard during this time. The inhibit will not be released until all characters of the bar code have been transmitted by the scanner. Once all characters of the bar code are transmitted, the bar code scanner releases the inhibit by placing the switching mechanism into a second state, so as to permit communications to take place between the keyboard and the PC. Any keyboard key that was typed in during transmission of the bar code will now be transmitted from the keyboard to the PC.

20 Claims, 16 Drawing Sheets



10/29/2003, EAST Version: 1.4.1

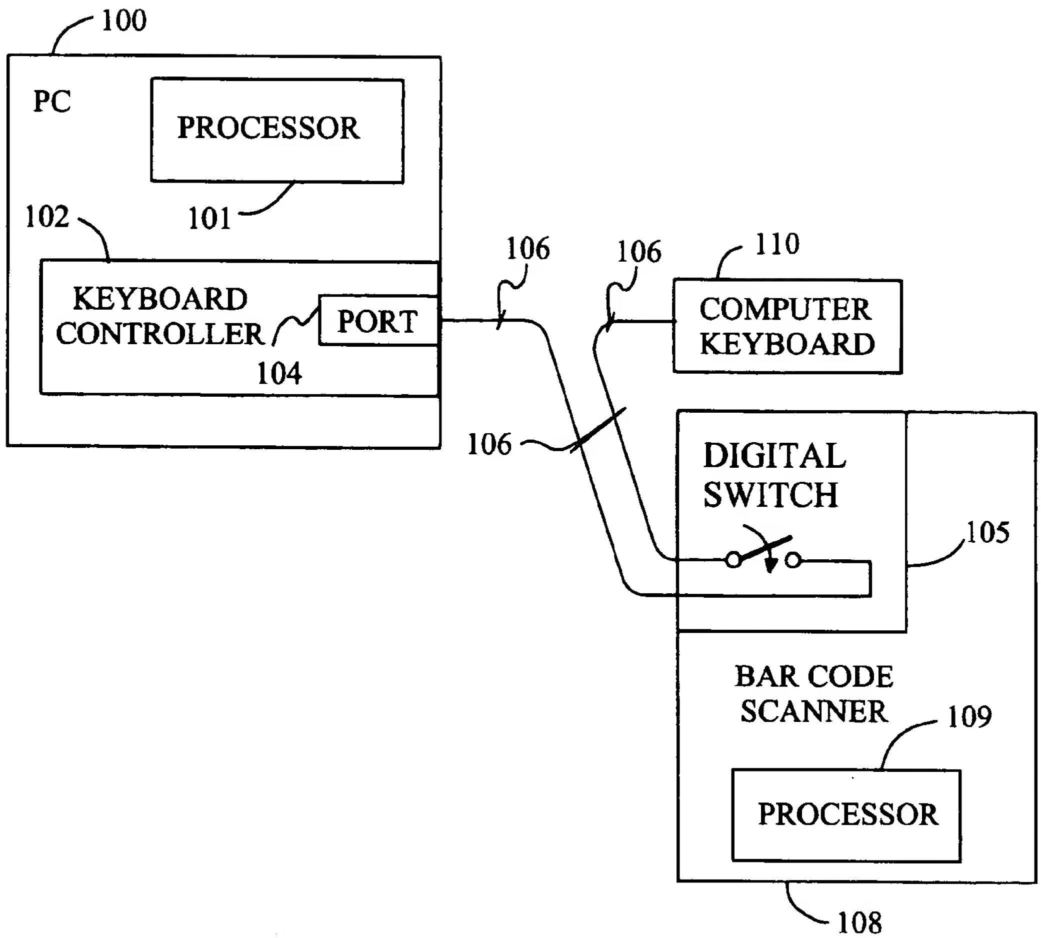


FIG. 1
PRIOR ART.

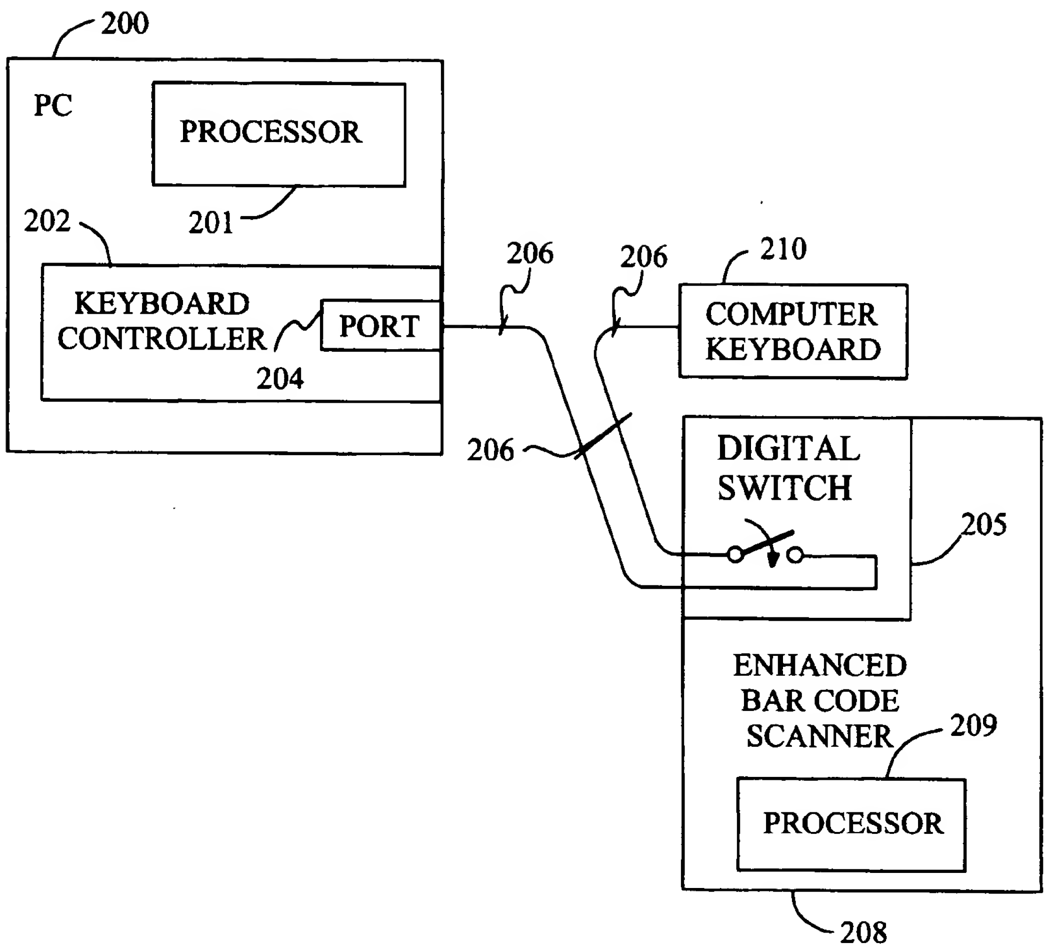

200
PC
PROCESSOR
201
202
KEYBOARD
CONTROLLER
PORT
204
206
206
210
COMPUTER
KEYBOARD
206
DIGITAL
SWITCH
205
ENHANCED
BAR CODE
SCANNER
209
PROCESSOR
208

FIG. 2

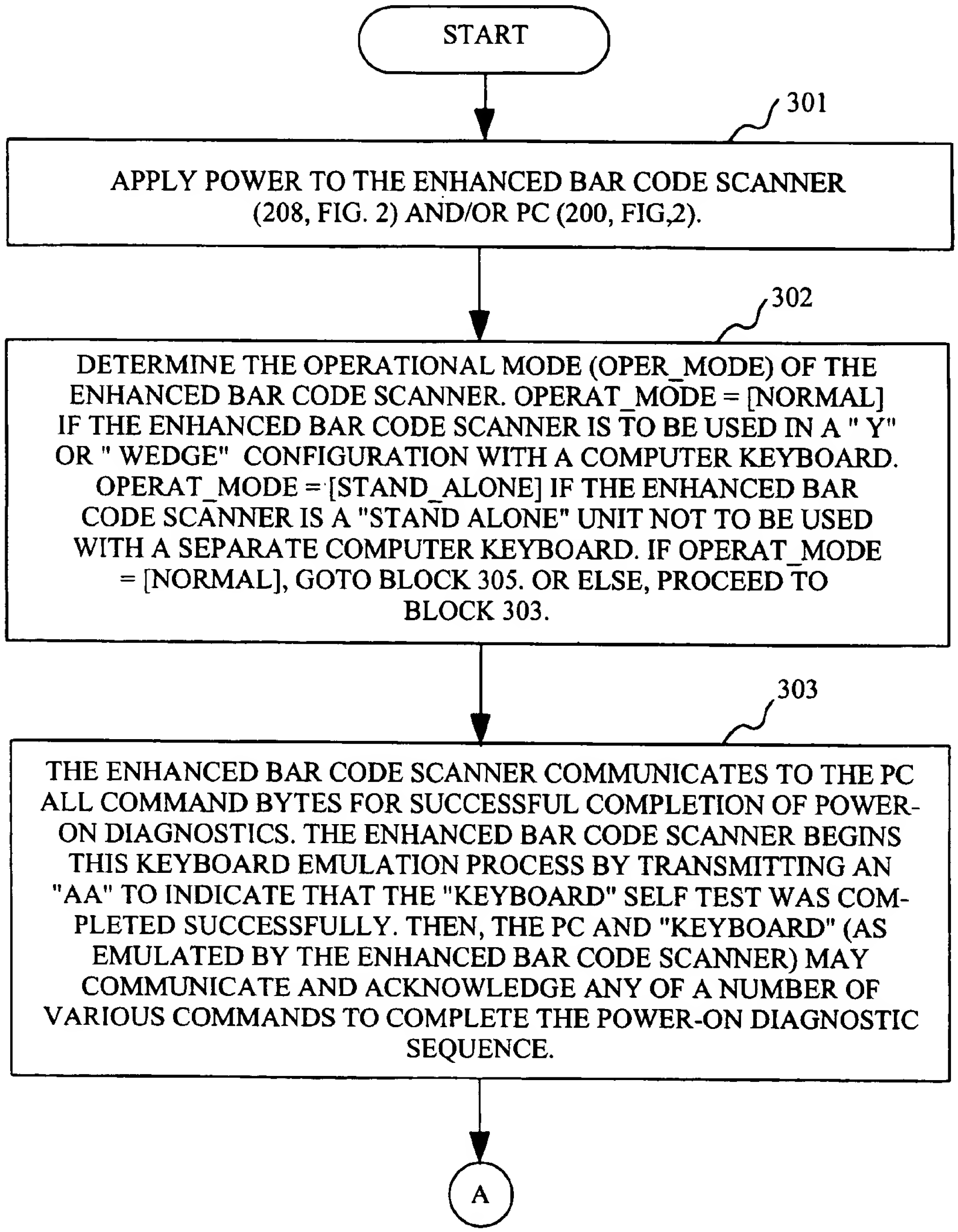

START
301
APPLY POWER TO THE ENHANCED BAR CODE SCANNER (208, FIG. 2) AND/OR PC (200, FIG,2).
302
DETERMINE THE OPERATIONAL MODE (OPER_MODE) OF THE ENHANCED BAR CODE SCANNER. OPERAT_MODE = [NORMAL] IF THE ENHANCED BAR CODE SCANNER IS TO BE USED IN A " Y" OR " WEDGE" CONFIGURATION WITH A COMPUTER KEYBOARD. OPERAT_MODE = [STAND_ALONE] IF THE ENHANCED BAR CODE SCANNER IS A "STAND ALONE" UNIT NOT TO BE USED WITH A SEPARATE COMPUTER KEYBOARD. IF OPERAT_MODE = [NORMAL], GOTO BLOCK 305. OR ELSE, PROCEED TO BLOCK 303.
303
THE ENHANCED BAR CODE SCANNER COMMUNICATES TO THE PC ALL COMMAND BYTES FOR SUCCESSFUL COMPLETION OF POWER-ON DIAGNOSTICS. THE ENHANCED BAR CODE SCANNER BEGINS THIS KEYBOARD EMULATION PROCESS BY TRANSMITTING AN "AA" TO INDICATE THAT THE "KEYBOARD" SELF TEST WAS COMPLETED SUCCESSFULLY. THEN, THE PC AND "KEYBOARD" (AS EMULATED BY THE ENHANCED BAR CODE SCANNER) MAY COMMUNICATE AND ACKNOWLEDGE ANY OF A NUMBER OF VARIOUS COMMANDS TO COMPLETE THE POWER-ON DIAGNOSTIC SEQUENCE.
A

FIG. 3A

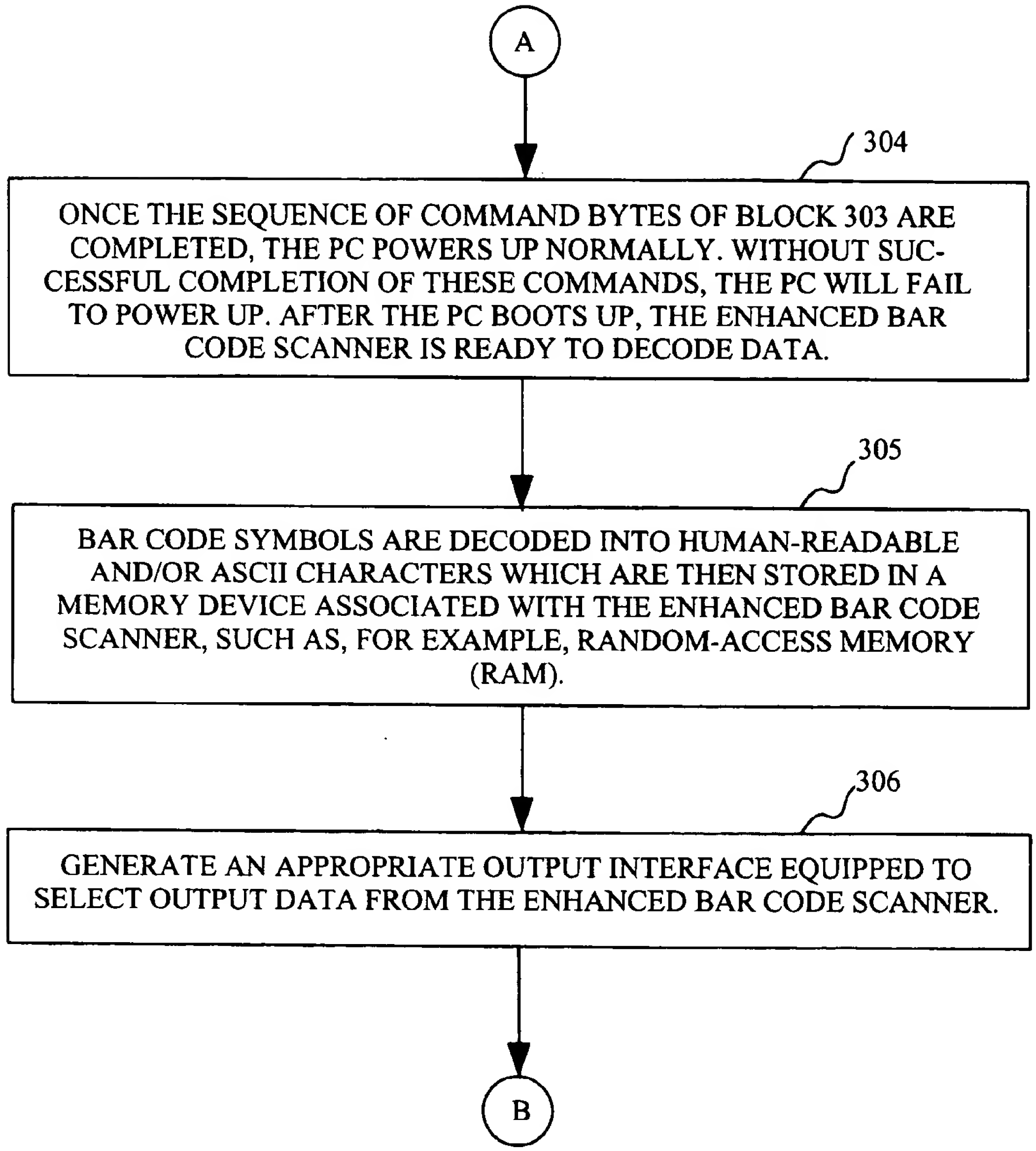

A
304
ONCE THE SEQUENCE OF COMMAND BYTES OF BLOCK 303 ARE COMPLETED, THE PC POWERS UP NORMALLY. WITHOUT SUCCESSFUL COMPLETION OF THESE COMMANDS, THE PC WILL FAIL TO POWER UP. AFTER THE PC BOOTS UP, THE ENHANCED BAR CODE SCANNER IS READY TO DECODE DATA.
305
BAR CODE SYMBOLS ARE DECODED INTO HUMAN-READABLE AND/OR ASCII CHARACTERS WHICH ARE THEN STORED IN A MEMORY DEVICE ASSOCIATED WITH THE ENHANCED BAR CODE SCANNER, SUCH AS, FOR EXAMPLE, RANDOM-ACCESS MEMORY (RAM).
306
GENERATE AN APPROPRIATE OUTPUT INTERFACE EQUIPPED TO SELECT OUTPUT DATA FROM THE ENHANCED BAR CODE SCANNER.
B

FIG. 3B

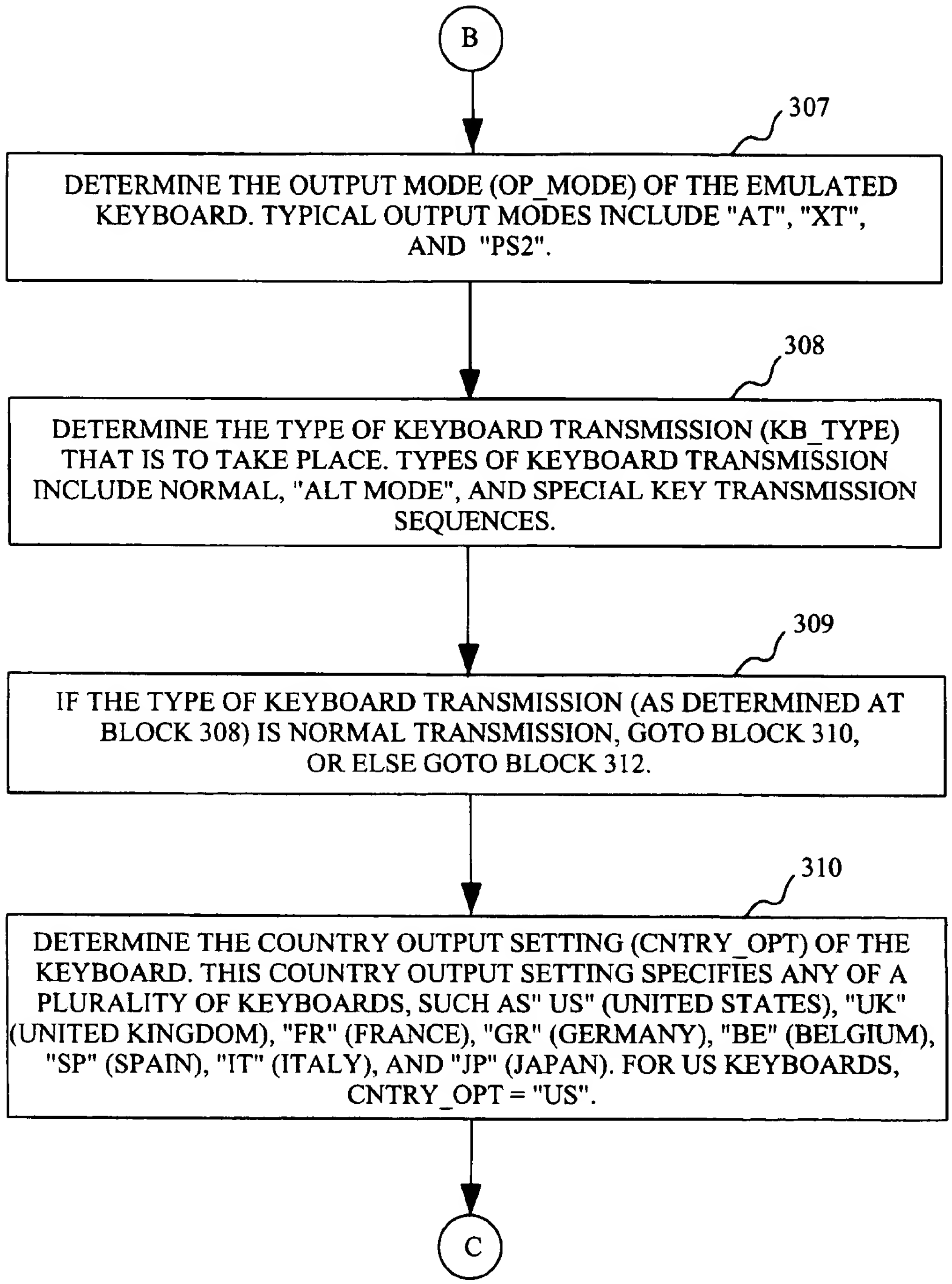

B
307
DETERMINE THE OUTPUT MODE (OP_MODE) OF THE EMULATED KEYBOARD. TYPICAL OUTPUT MODES INCLUDE "AT", "XT", AND "PS2".
308
DETERMINE THE TYPE OF KEYBOARD TRANSMISSION (KB_TYPE) THAT IS TO TAKE PLACE. TYPES OF KEYBOARD TRANSMISSION INCLUDE NORMAL, "ALT MODE", AND SPECIAL KEY TRANSMISSION SEQUENCES.
309
IF THE TYPE OF KEYBOARD TRANSMISSION (AS DETERMINED AT BLOCK 308) IS NORMAL TRANSMISSION, GOTO BLOCK 310, OR ELSE GOTO BLOCK 312.
310
DETERMINE THE COUNTRY OUTPUT SETTING (CNTRY_OPT) OF THE KEYBOARD. THIS COUNTRY OUTPUT SETTING SPECIFIES ANY OF A PLURALITY OF KEYBOARDS, SUCH AS" US" (UNITED STATES), "UK" (UNITED KINGDOM), "FR" (FRANCE), "GR" (GERMANY), "BE" (BELGIUM), "SP" (SPAIN), "IT" (ITALY), AND "JP" (JAPAN). FOR US KEYBOARDS, CNTRY_OPT = "US".
C

FIG. 3C

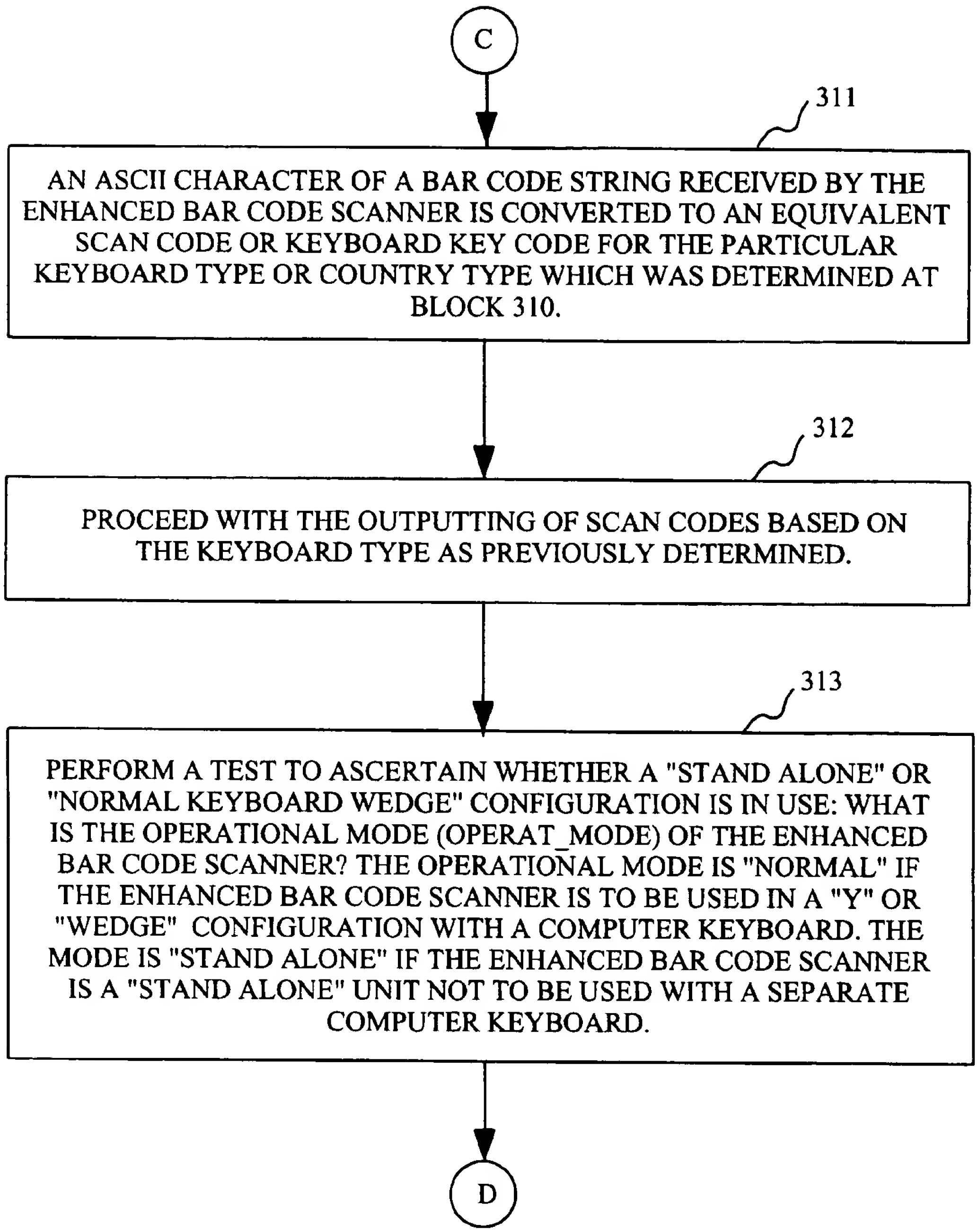

C
311
AN ASCII CHARACTER OF A BAR CODE STRING RECEIVED BY THE ENHANCED BAR CODE SCANNER IS CONVERTED TO AN EQUIVALENT SCAN CODE OR KEYBOARD KEY CODE FOR THE PARTICULAR KEYBOARD TYPE OR COUNTRY TYPE WHICH WAS DETERMINED AT BLOCK 310.
312
PROCEED WITH THE OUTPUTTING OF SCAN CODES BASED ON THE KEYBOARD TYPE AS PREVIOUSLY DETERMINED.
313
PERFORM A TEST TO ASCERTAIN WHETHER A "STAND ALONE" OR "NORMAL KEYBOARD WEDGE" CONFIGURATION IS IN USE: WHAT IS THE OPERATIONAL MODE (OPERAT_MODE) OF THE ENHANCED BAR CODE SCANNER? THE OPERATIONAL MODE IS "NORMAL" IF THE ENHANCED BAR CODE SCANNER IS TO BE USED IN A "Y" OR "WEDGE" CONFIGURATION WITH A COMPUTER KEYBOARD. THE MODE IS "STAND ALONE" IF THE ENHANCED BAR CODE SCANNER IS A "STAND ALONE" UNIT NOT TO BE USED WITH A SEPARATE COMPUTER KEYBOARD.
D

FIG. 3D

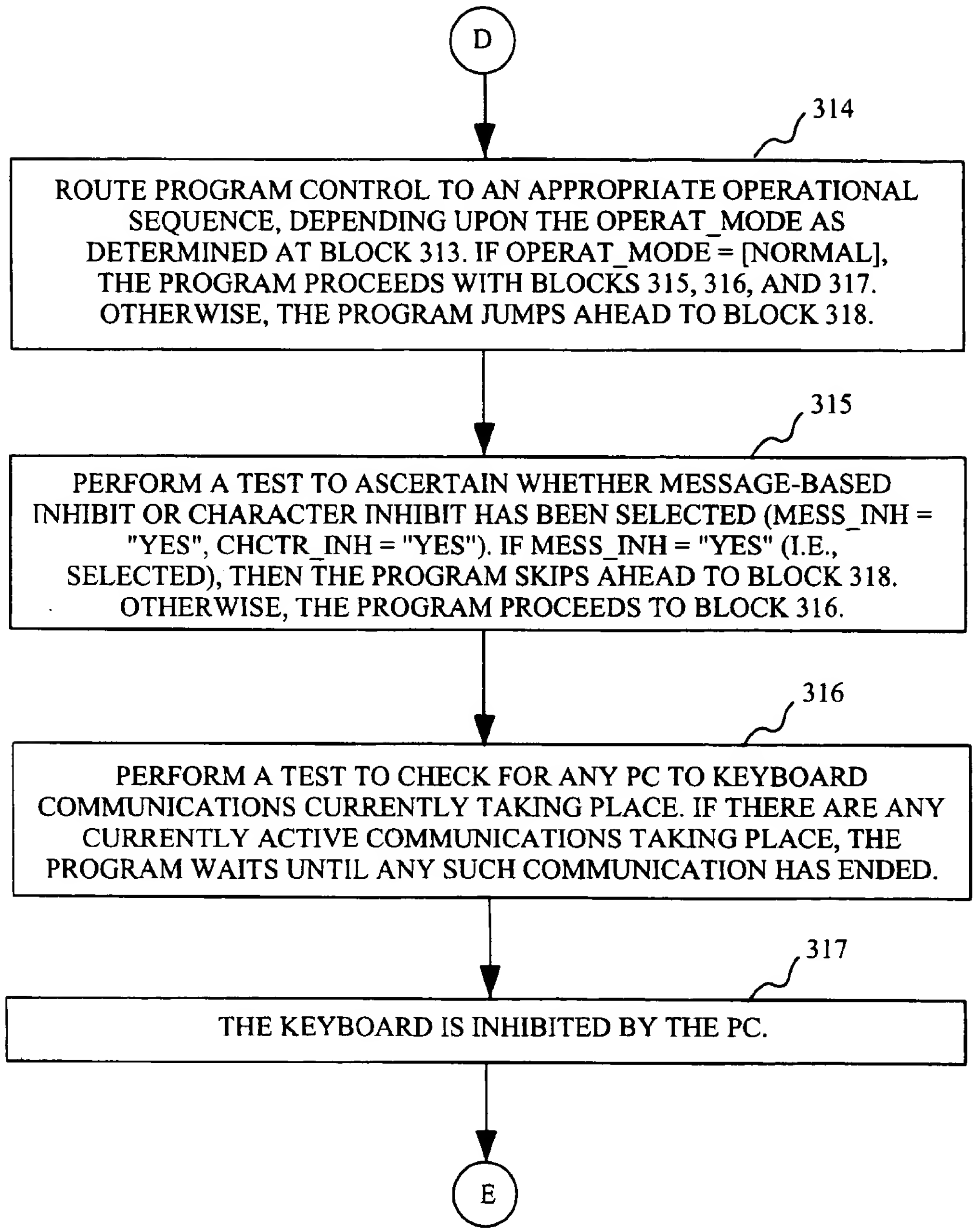

D
314
ROUTE PROGRAM CONTROL TO AN APPROPRIATE OPERATIONAL SEQUENCE, DEPENDING UPON THE OPERAT_MODE AS DETERMINED AT BLOCK 313. IF OPERAT_MODE = [NORMAL], THE PROGRAM PROCEEDS WITH BLOCKS 315, 316, AND 317. OTHERWISE, THE PROGRAM JUMPS AHEAD TO BLOCK 318.
315
PERFORM A TEST TO ASCERTAIN WHETHER MESSAGE-BASED INHIBIT OR CHARACTER INHIBIT HAS BEEN SELECTED (MESS_INH = "YES", CHCTR_INH = "YES"). IF MESS_INH = "YES" (I.E., SELECTED), THEN THE PROGRAM SKIPS AHEAD TO BLOCK 318. OTHERWISE, THE PROGRAM PROCEEDS TO BLOCK 316.
316
PERFORM A TEST TO CHECK FOR ANY PC TO KEYBOARD COMMUNICATIONS CURRENTLY TAKING PLACE. IF THERE ARE ANY CURRENTLY ACTIVE COMMUNICATIONS TAKING PLACE, THE PROGRAM WAITS UNTIL ANY SUCH COMMUNICATION HAS ENDED.
317
THE KEYBOARD IS INHIBITED BY THE PC.
E

FIG. 3E

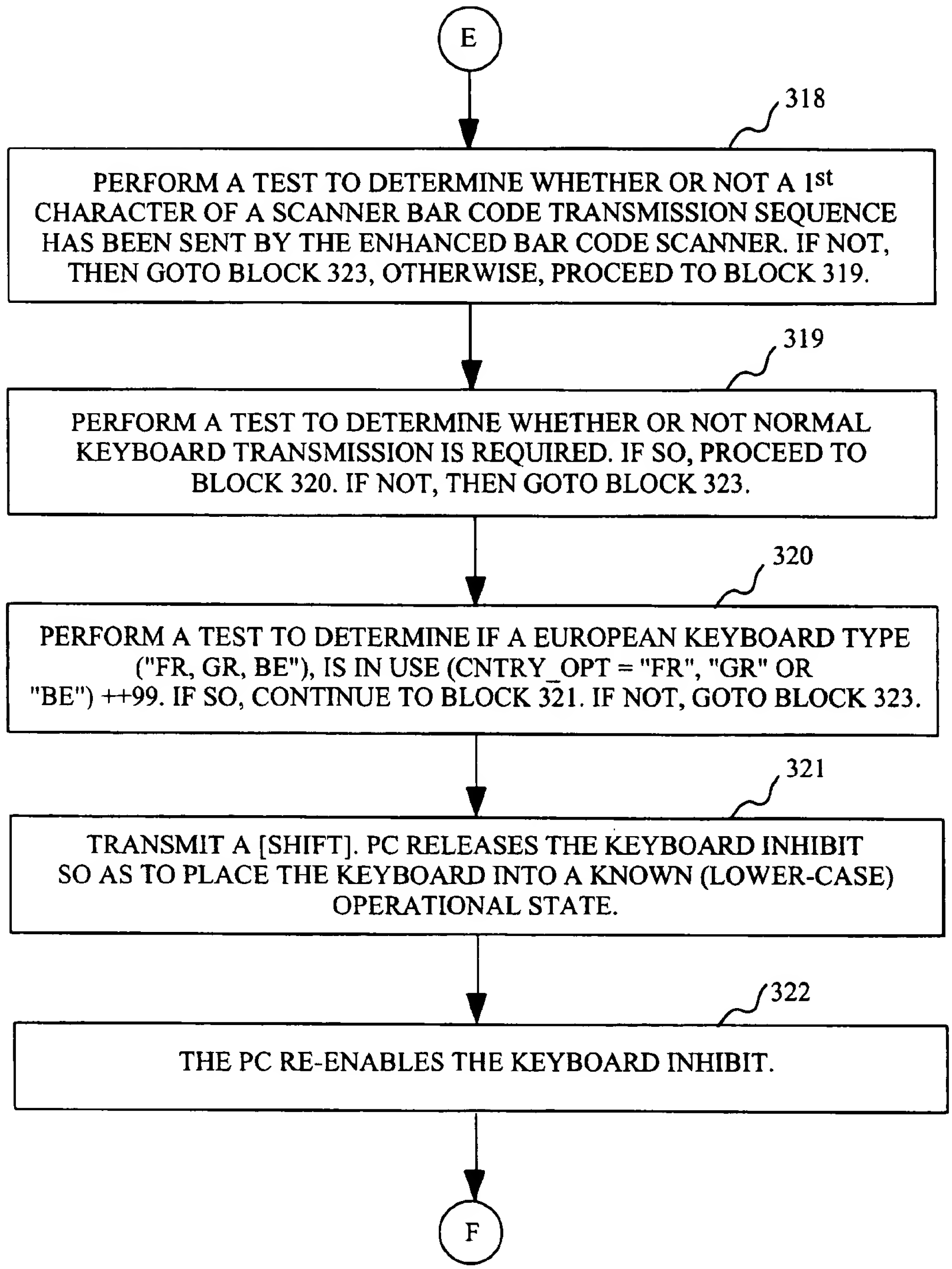

E
318
PERFORM A TEST TO DETERMINE WHETHER OR NOT A 1st CHARACTER OF A SCANNER BAR CODE TRANSMISSION SEQUENCE HAS BEEN SENT BY THE ENHANCED BAR CODE SCANNER. IF NOT, THEN GOTO BLOCK 323, OTHERWISE, PROCEED TO BLOCK 319.
319
PERFORM A TEST TO DETERMINE WHETHER OR NOT NORMAL KEYBOARD TRANSMISSION IS REQUIRED. IF SO, PROCEED TO BLOCK 320. IF NOT, THEN GOTO BLOCK 323.
320
PERFORM A TEST TO DETERMINE IF A EUROPEAN KEYBOARD TYPE ("FR, GR, BE"), IS IN USE (CNTRY_OPT = "FR", "GR" OR "BE") ++99. IF SO, CONTINUE TO BLOCK 321. IF NOT, GOTO BLOCK 323.
321
TRANSMIT A [SHIFT]. PC RELEASES THE KEYBOARD INHIBIT SO AS TO PLACE THE KEYBOARD INTO A KNOWN (LOWER-CASE) OPERATIONAL STATE.
322
THE PC RE-ENABLES THE KEYBOARD INHIBIT.
F

FIG. 3F

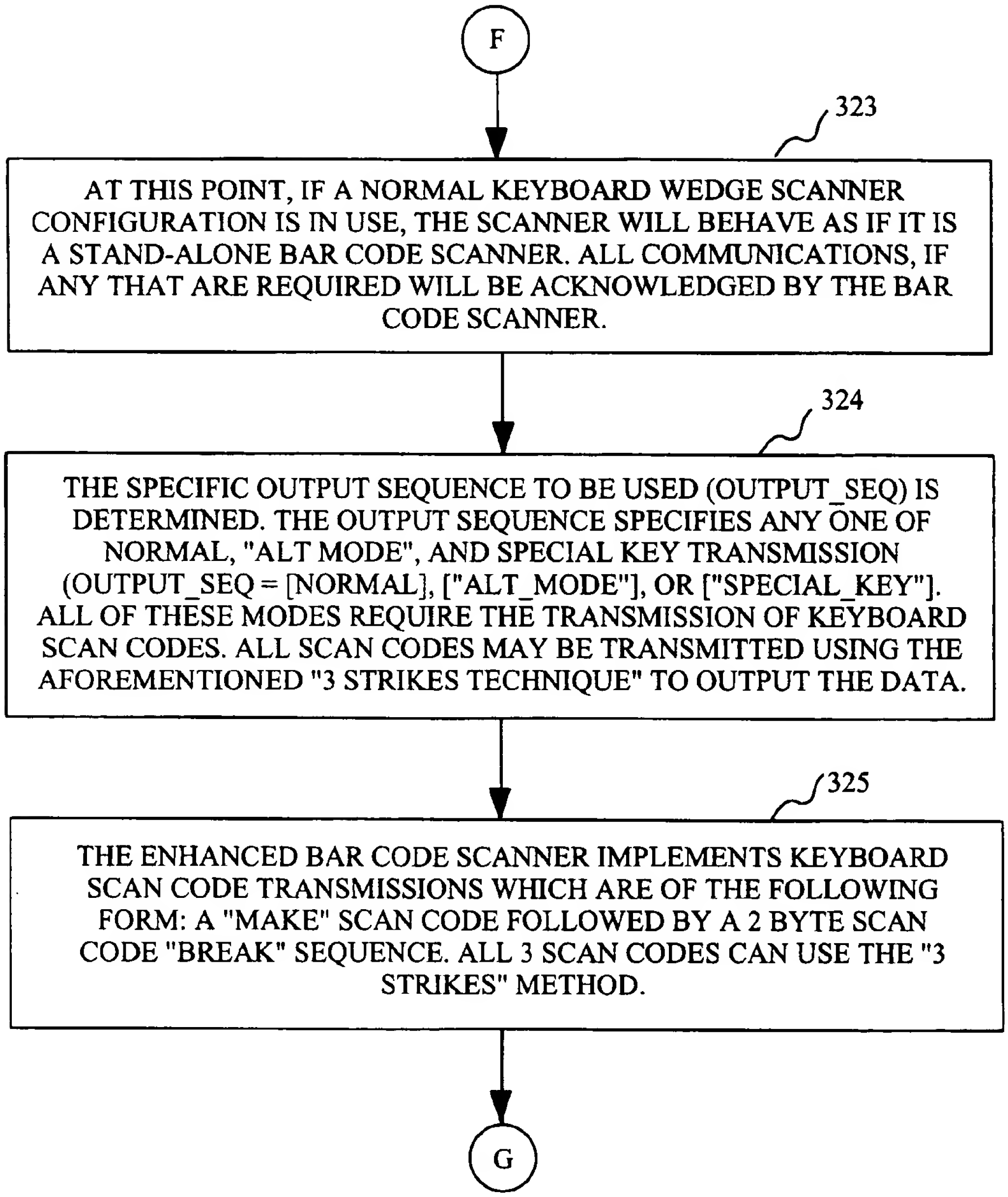

F
323
AT THIS POINT, IF A NORMAL KEYBOARD WEDGE SCANNER CONFIGURATION IS IN USE, THE SCANNER WILL BEHAVE AS IF IT IS A STAND-ALONE BAR CODE SCANNER. ALL COMMUNICATIONS, IF ANY THAT ARE REQUIRED WILL BE ACKNOWLEDGED BY THE BAR CODE SCANNER.
324
THE SPECIFIC OUTPUT SEQUENCE TO BE USED (OUTPUT_SEQ) IS DETERMINED. THE OUTPUT SEQUENCE SPECIFIES ANY ONE OF NORMAL, "ALT MODE", AND SPECIAL KEY TRANSMISSION (OUTPUT_SEQ = [NORMAL], ["ALT_MODE"], OR ["SPECIAL_KEY"]. ALL OF THESE MODES REQUIRE THE TRANSMISSION OF KEYBOARD SCAN CODES. ALL SCAN CODES MAY BE TRANSMITTED USING THE AFOREMENTIONED "3 STRIKES TECHNIQUE" TO OUTPUT THE DATA.
325
THE ENHANCED BAR CODE SCANNER IMPLEMENTS KEYBOARD SCAN CODE TRANSMISSIONS WHICH ARE OF THE FOLLOWING FORM: A "MAKE" SCAN CODE FOLLOWED BY A 2 BYTE SCAN CODE "BREAK" SEQUENCE. ALL 3 SCAN CODES CAN USE THE "3 STRIKES" METHOD.
G

FIG. 3G

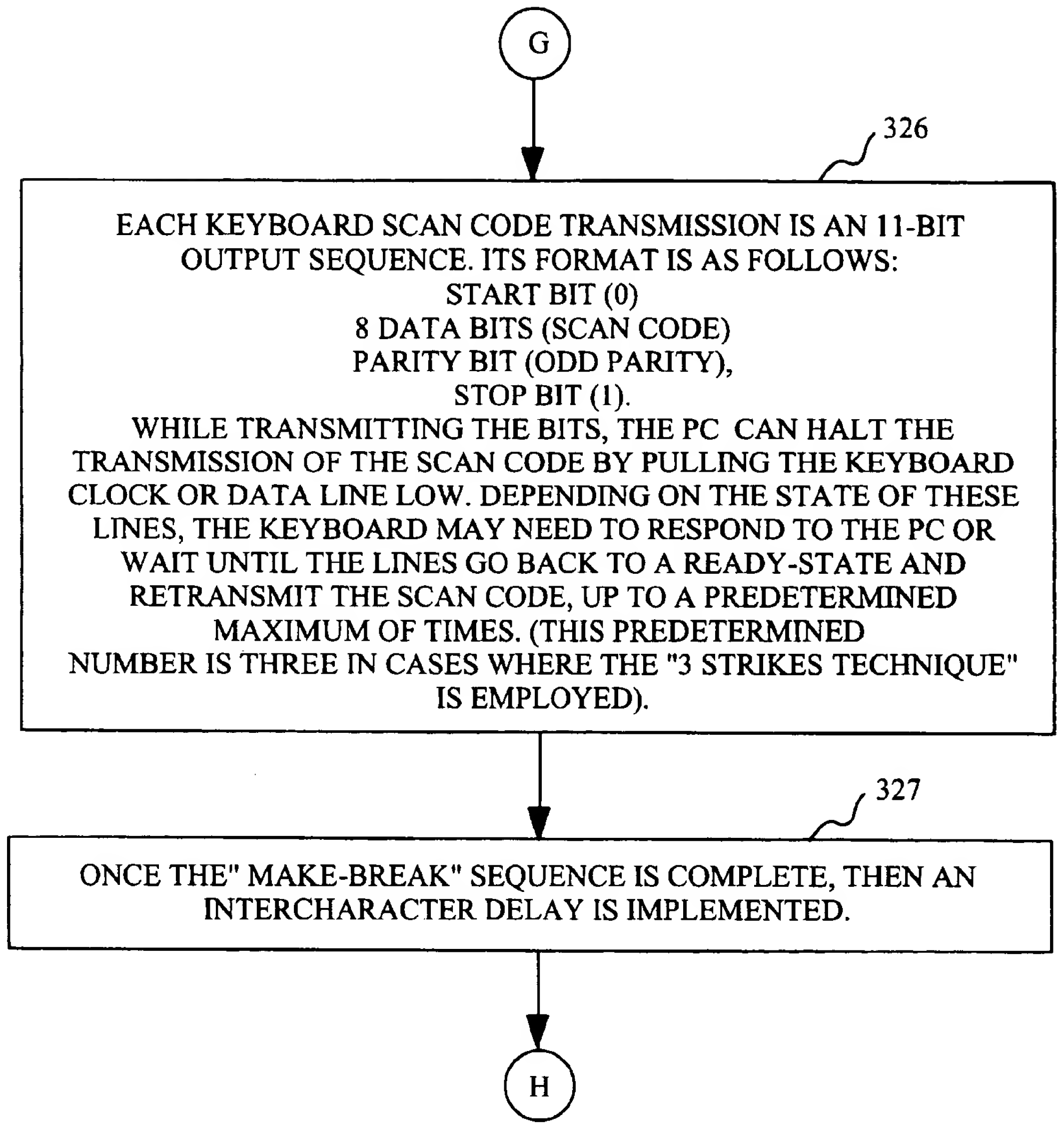

G
326
EACH KEYBOARD SCAN CODE TRANSMISSION IS AN 11-BIT OUTPUT SEQUENCE. ITS FORMAT IS AS FOLLOWS:
START BIT (0)
8 DATA BITS (SCAN CODE)
PARITY BIT (ODD PARITY),
STOP BIT (1).
WHILE TRANSMITTING THE BITS, THE PC CAN HALT THE TRANSMISSION OF THE SCAN CODE BY PULLING THE KEYBOARD CLOCK OR DATA LINE LOW. DEPENDING ON THE STATE OF THESE LINES, THE KEYBOARD MAY NEED TO RESPOND TO THE PC OR WAIT UNTIL THE LINES GO BACK TO A READY-STATE AND RETRANSMIT THE SCAN CODE, UP TO A PREDETERMINED MAXIMUM OF TIMES. (THIS PREDETERMINED NUMBER IS THREE IN CASES WHERE THE "3 STRIKES TECHNIQUE" IS EMPLOYED).
327
ONCE THE" MAKE-BREAK" SEQUENCE IS COMPLETE, THEN AN INTERCHARACTER DELAY IS IMPLEMENTED.
H

FIG. 3H

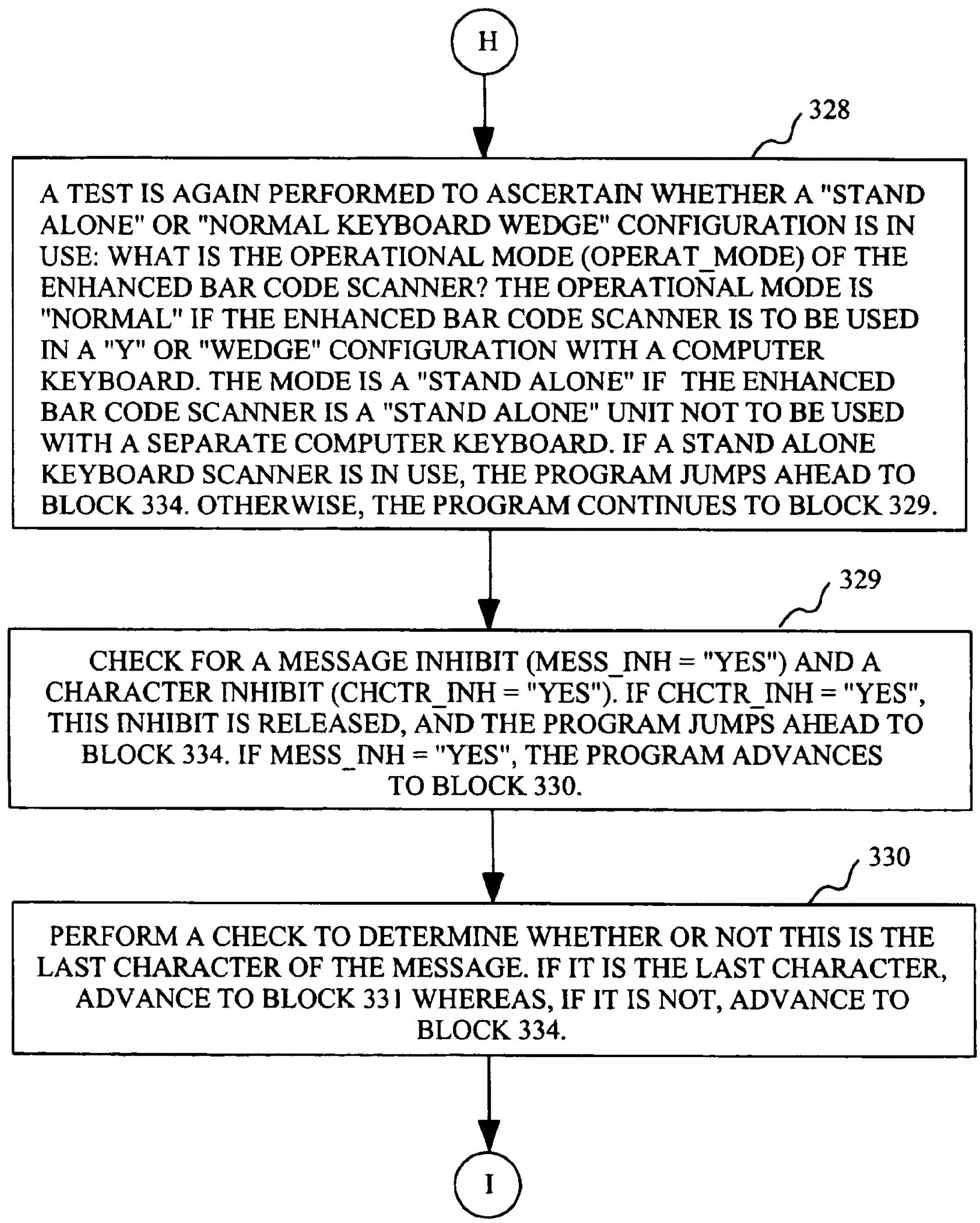

H
328
A TEST IS AGAIN PERFORMED TO ASCERTAIN WHETHER A "STAND ALONE" OR "NORMAL KEYBOARD WEDGE" CONFIGURATION IS IN USE: WHAT IS THE OPERATIONAL MODE (OPERAT_MODE) OF THE ENHANCED BAR CODE SCANNER? THE OPERATIONAL MODE IS "NORMAL" IF THE ENHANCED BAR CODE SCANNER IS TO BE USED IN A "Y" OR "WEDGE" CONFIGURATION WITH A COMPUTER KEYBOARD. THE MODE IS A "STAND ALONE" IF THE ENHANCED BAR CODE SCANNER IS A "STAND ALONE" UNIT NOT TO BE USED WITH A SEPARATE COMPUTER KEYBOARD. IF A STAND ALONE KEYBOARD SCANNER IS IN USE, THE PROGRAM JUMPS AHEAD TO BLOCK 334. OTHERWISE, THE PROGRAM CONTINUES TO BLOCK 329.
329
CHECK FOR A MESSAGE INHIBIT (MESS_INH = "YES") AND A CHARACTER INHIBIT (CHCTR_INH = "YES"). IF CHCTR_INH = "YES", THIS INHIBIT IS RELEASED, AND THE PROGRAM JUMPS AHEAD TO BLOCK 334. IF MESS_INH = "YES", THE PROGRAM ADVANCES TO BLOCK 330.
330
PERFORM A CHECK TO DETERMINE WHETHER OR NOT THIS IS THE LAST CHARACTER OF THE MESSAGE. IF IT IS THE LAST CHARACTER, ADVANCE TO BLOCK 331 WHEREAS, IF IT IS NOT, ADVANCE TO BLOCK 334.
I

FIG. 3I

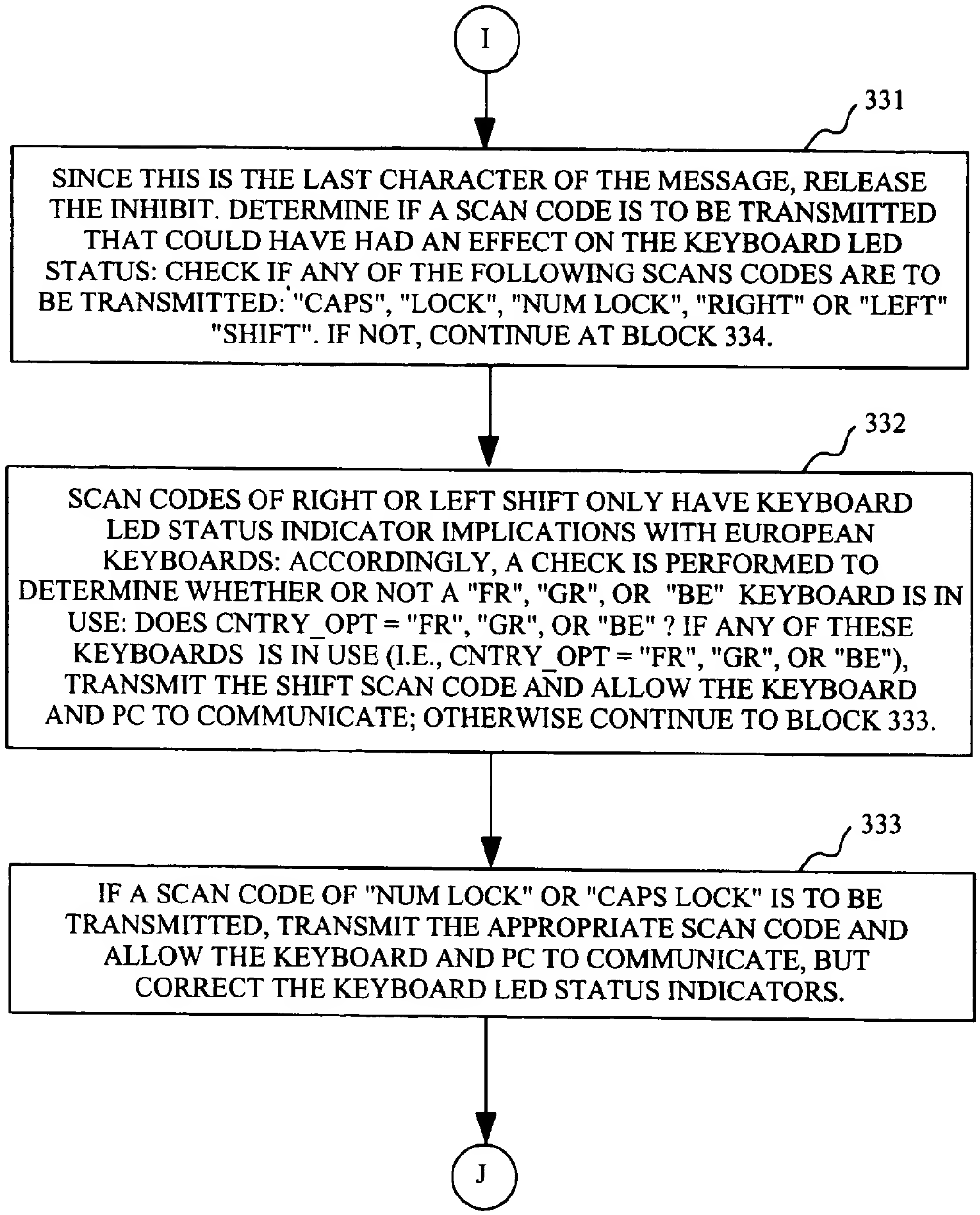


FIG. 3J

10/29/2003, EAST Version: 1.4.1

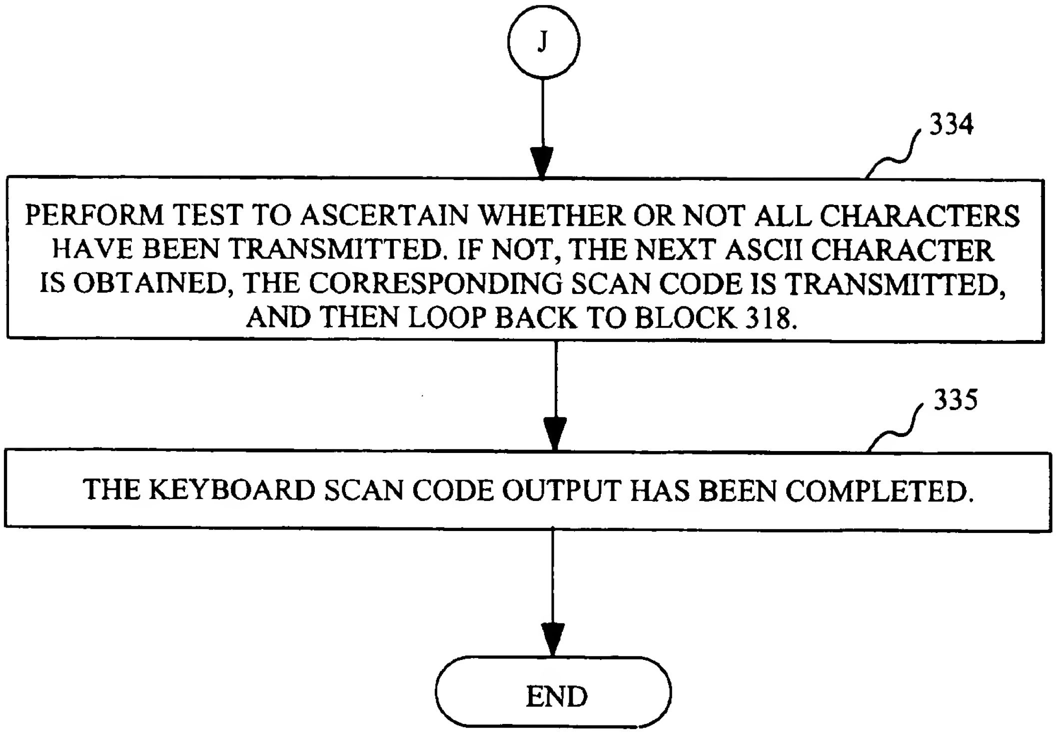

J
334
PERFORM TEST TO ASCERTAIN WHETHER OR NOT ALL CHARACTERS HAVE BEEN TRANSMITTED. IF NOT, THE NEXT ASCII CHARACTER IS OBTAINED, THE CORRESPONDING SCAN CODE IS TRANSMITTED, AND THEN LOOP BACK TO BLOCK 318.
335
THE KEYBOARD SCAN CODE OUTPUT HAS BEEN COMPLETED.
END

FIG. 3K

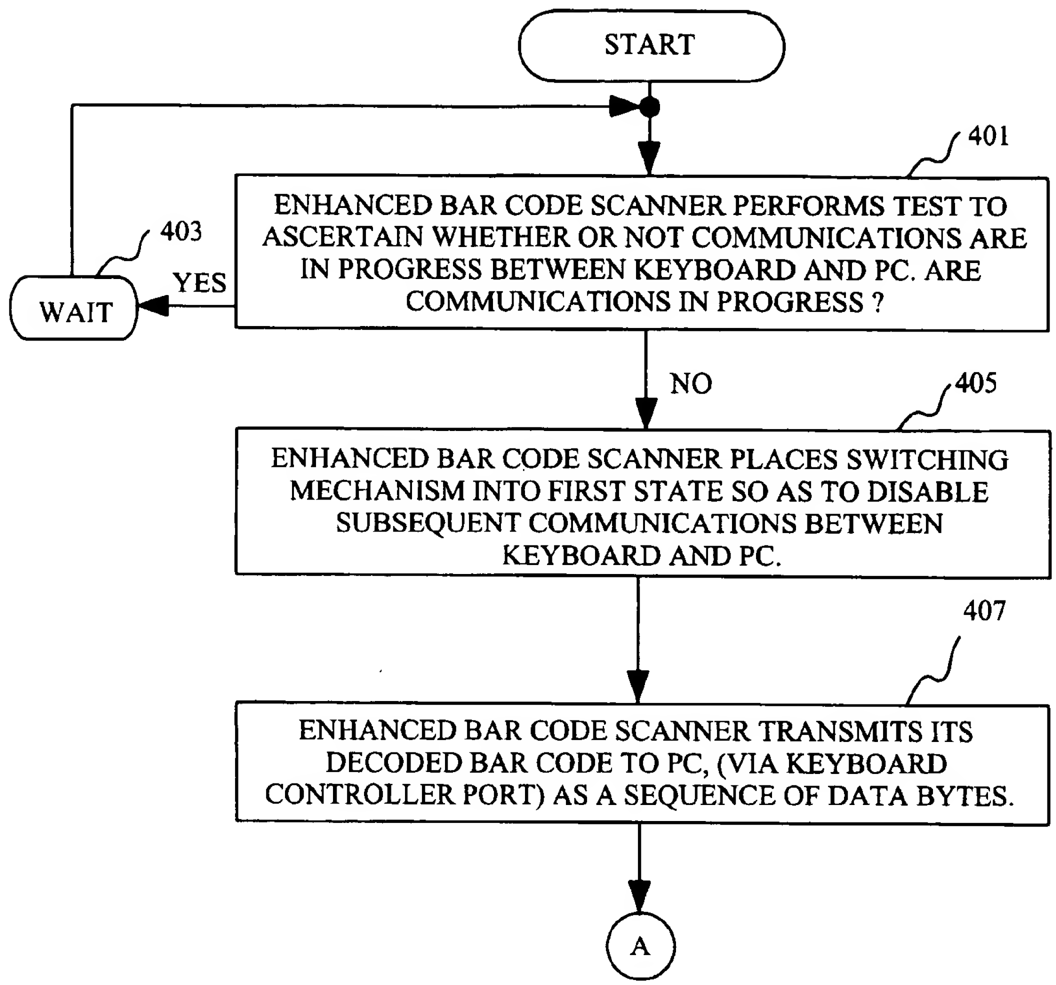

START
401
ENHANCED BAR CODE SCANNER PERFORMS TEST TO ASCERTAIN WHETHER OR NOT COMMUNICATIONS ARE IN PROGRESS BETWEEN KEYBOARD AND PC. ARE COMMUNICATIONS IN PROGRESS ?
403
YES
WAIT
NO
405
ENHANCED BAR CODE SCANNER PLACES SWITCHING MECHANISM INTO FIRST STATE SO AS TO DISABLE SUBSEQUENT COMMUNICATIONS BETWEEN KEYBOARD AND PC.
407
ENHANCED BAR CODE SCANNER TRANSMITS ITS DECODED BAR CODE TO PC, (VIA KEYBOARD CONTROLLER PORT) AS A SEQUENCE OF DATA BYTES.
A

FIG. 4A

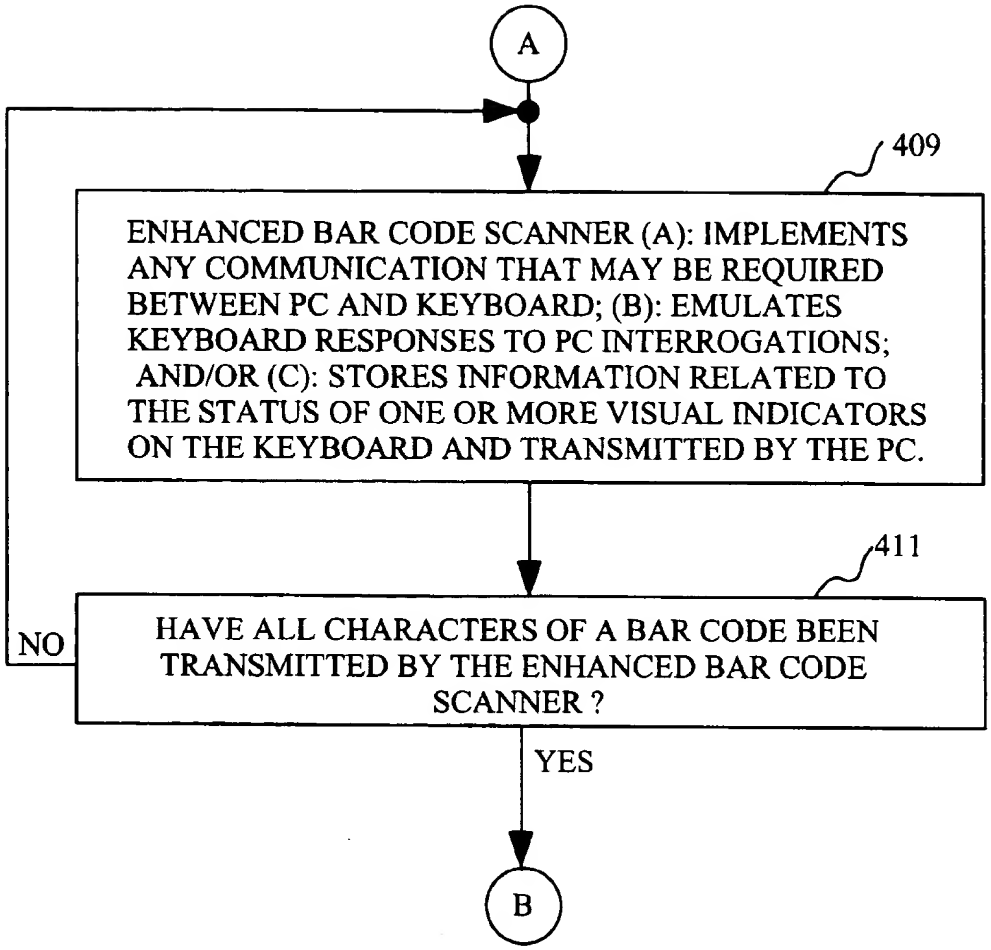

A
409
ENHANCED BAR CODE SCANNER (A): IMPLEMENTS ANY COMMUNICATION THAT MAY BE REQUIRED BETWEEN PC AND KEYBOARD; (B): EMULATES KEYBOARD RESPONSES TO PC INTERROGATIONS; AND/OR (C): STORES INFORMATION RELATED TO THE STATUS OF ONE OR MORE VISUAL INDICATORS ON THE KEYBOARD AND TRANSMITTED BY THE PC.
411
HAVE ALL CHARACTERS OF A BAR CODE BEEN TRANSMITTED BY THE ENHANCED BAR CODE SCANNER ?
NO
YES
B

FIG. 4B

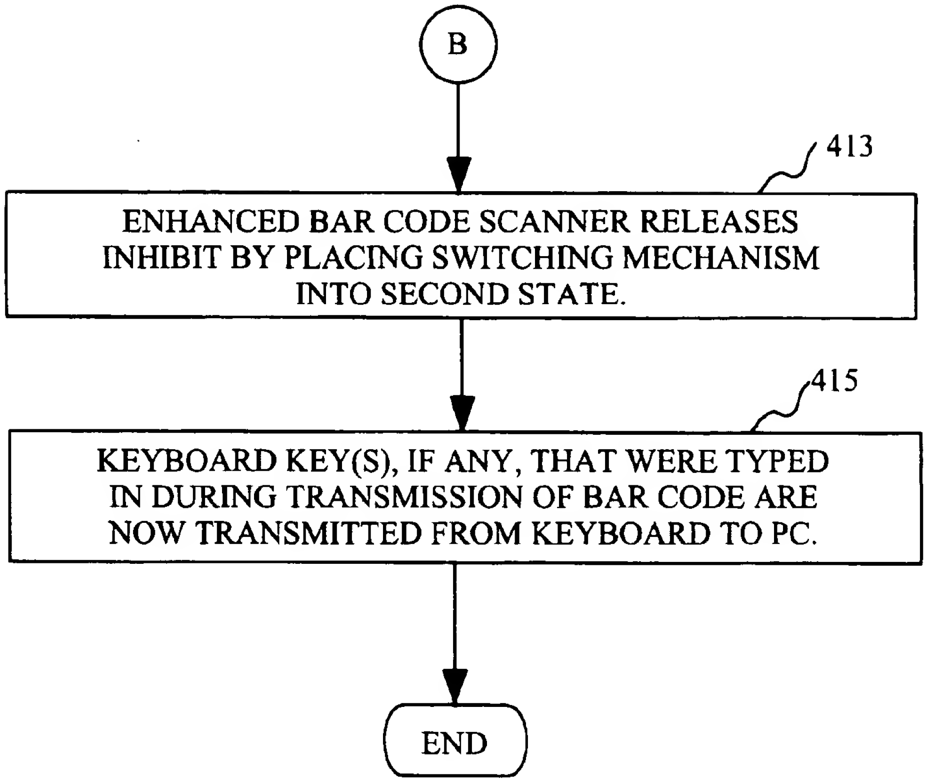

B
413
ENHANCED BAR CODE SCANNER RELEASES INHIBIT BY PLACING SWITCHING MECHANISM INTO SECOND STATE.
415
KEYBOARD KEY(S), IF ANY, THAT WERE TYPED IN DURING TRANSMISSION OF BAR CODE ARE NOW TRANSMITTED FROM KEYBOARD TO PC.
END

FIG. 4C

TECHNIQUES FOR INTERFACING A BAR CODE SCANNER TO A PC USING A MESSAGE-BASED AND/OR CHARACTER-BASED KEYBOARD INHIBIT

The present patent application is based upon Provisional Application Serial No. 60/144,389, filed on Jul. 16^{th}, 1999, and entitled, "Techniques for Interfacing a Bar Code Scanner to a PC Using A Message-Based and/or Character-Based Keyboard Inhibit", the disclosure of which is incorporated herein by reference.

FIELD OF THE INVENTION

To The invention relates generally to bar code scanners, and more specifically, to techniques for interfacing bar code scanners with computers.

BACKGROUND OF THE INVENTION

For years, various bar code scanner manufacturers have been selling keyboard-wedge bar code scanners. With reference to FIG. **1**, bar code scanner **108** is connected to a personal computer (PC) **100** keyboard controller port **104** and a computer keyboard **110** in an Y or wedge type configuration. Bar code scanner **108** may contain an on-board processor **109**. PC **100** also contains a processor **101**. The Y or wedge type configuration is implemented using interconnection cable **106**. A first section of interconnection cable connects keyboard controller port **104** to bar code scanner **108**, and a second section of interconnection cable connects bar code scanner **108** to computer keyboard **110**.

From a conceptual standpoint, imagine that a digital switch **105** (within bar code scanner **108**) is set up to selectively connect and disconnect the first section of interconnection cable **106** to the second section of interconnection cable **106**. When the digital switch **105** is in a closed state, the keyboard controller port **104** is connected to bar code scanner **108** and computer keyboard **110** via the first and second sections of interconnection cable **106**. When the digital switch **105** is in an open state, the keyboard controller port **104** is only connected to the bar code scanner **108**, and the keyboard **110** is isolated from the bar code scanner **108** as well as the keyboard controller port **104**.

When digital switch **105** is in the closed state, computer keyboard **110** and bar code scanner **108** are placed in parallel across controller port **104**. This parallel configuration is used because keyboard controller **102** circuitry within presently-existing PCs (personal computers) **100** and laptop computers attempts to detect the existence of a computer keyboard **110** connected to the keyboard controller port **104**. The controller port **104** needs to be connected to a computer keyboard **110**, even if the keyboard is not to be used for subsequent data entry, and even if the controller port **104** is also connected to an input device other than a computer keyboard. If the keyboard controller **102** circuitry does not detect a keyboard connected to the controller port **104**, the PC **100** and/or laptop may then disable the port, preventing any further inputting of data. In the operational environment of FIG. **1**, this disablement poses a problem, because we desire to input further data as bar codes are detected and decoded.

In actuality, the use of a digital switch **105** to completely isolate the keyboard from the keyboard controller port is an over-simplification, presented for purposes of conceptual expediency. Real-world systems may isolate the keyboard using any of a variety of techniques. However, a common approach is to bring the keyboard clock line to a logic "low" state, while allowing the keyboard data line to float "high". In practice, connections between the LED status indicator lamps (num-lock, caps-lock, and scroll-lock) and the keyboard controller port may remain in place, even while the keyboard is inhibited.

With the increasing use of laptop computers and keyless data entry, the keyboard controller port shows great potential as a convenient, somewhat standardized, and readily available data input channel. However, this potential could be advantageously exploited only if it were possible to find some way around the necessity of connecting this port to an actual computer keyboard. By way of clarification, there are a number of existing programs that do not require the use of a computer keyboard per se, but these programs have neglected to provide mechanisms by which a computer keyboard is emulated, so as to prevent the controller port from being disabled.

Assume that a conventional keyboard wedge bar code scanner is connected to a keyboard controller port of a PC or laptop, as shown in FIG. **1**, while, at the same time, the keyboard that is connected in the parallel (Y) configuration is eliminated. Will the hardware configuration of FIG. **1** still function as desired? It is important to realize that keyboard-to-PC communications is implemented by means of a 2-way channel. Other types of data must be communicated between the PC and the keyboard in addition to information specifying the key or keys that were pressed. When a PC is powered up, the PC is programmed to check for the existence of a primary data input device, which is typically a keyboard. The PC begins a data exchange with the keyboard, and this communication is called "power-on diagnostics". If the keyboard is not present, or if the power-on diagnostics fail, the PC will not boot up. Accordingly, if a normal boot-up is desired, the keyboard shown in FIG. **1** should not be eliminated. In the case of laptop computers, a similar situation exists. A communication protocol is used to sense the presence of an external keyboard that is connected to the laptop's external keyboard port. If the laptop computer fails to detect a keyboard at the external keyboard port, then the laptop computer may disable its external keyboard port.

Even if a technique were to be developed by which a bar code scanner could successfully emulate a computer keyboard during boot-up, another problem would then arise. Eleven (11)-bit transmission protocols are utilized almost universally to provide keyboard to PC data transfer. The transmission protocol begins with a Start bit (low), followed by 8 data bits (these bits represent a scan code which, in turn, could be used to represent a portion of a detected bar code or a key press), a Parity bit (odd parity) and finally a Stop bit (high). Three of these 11-bit "bytes" are used to represent a "character". Problems arise because this protocol allows the PC (or laptop) to interrupt the transmitted sequence up through the 9^{th} bit.

If the keyboard **110** or keyboard wedge scanner (i.e., bar code scanner **108**) begins transmitting a data byte, the PC can prevent the successful communication of that byte, up to and including the 9^{th} th transmitted bit. To make matters worse, a typical bar code includes a sequence of several 3-byte characters. An interruption at any time during transmission of that sequence will result in loss or corruption of the entire bar code. This keyboard inhibit problem is described in greater detail in a reference book entitled, "PC KEYBOARD DESIGN", by J. Konzak, and published by Annabook.

With the advent of multitasking operating systems, sophisticated network operating systems, and dual-keyboard

10/29/2003, EAST Version: 1.4.1

port laptops and PC's, the problem worsens. Some of these operating systems interrupt keyboard port communications on a frequent and periodic basis, such as once every ten milliseconds. Of course, in operational environments where the keyboard controller port is not used with an auxiliary input devices, the computer keyboard will sense this stoppage of communications and retransmit the scan code after the PC releases its halt or inhibit of the keyboard. Existing bar code scanners are not so equipped. If a data transmission from a bar code scanner is interrupted, the scanner has no mechanism by which to ascertain whether or not a data entry error has occurred.

Existing scanners do not monitor for the existence of an inhibit signal, so the scanner cannot infer a data entry error based upon the existence of an inhibit. Pursuant to prior art techniques, bar code scanners had not monitored the inhibit signal because the decoded bar codes were wedged into the transmitted data for brief periods of time, relative to typed-in keyboard data. Also, as a practical matter, the keyboard BIOS virtually never inhibits the keyboard during scan code transmission. Any problems that may have been encountered were handled by changing certain programmable parameters such as inter-character delays or inter-scan-code delays.

In some existing wedge scanner systems, PC **100** maintains control over keyboard control port communications. After all, the PC is equipped to inhibit keyboard transmission at any time by sending out a "keyboard inhibit" signal. However, a bar code scanner **108** could also be programmed to inhibit the keyboard, and some presently-existing bar code scanners are programmed take advantage of this fact.

Various keyboard wedge bar code scanners **108** have been programmed to transmit decoded bar code data on a character-by-character basis. This approach operates as follows: Test for active communications between the keyboard **110** and the PC **100**. If no communication is detected, the bar code scanner **108** opens digital switch **105**, thereby inhibiting the keyboard **110**. The bar code scanner **108** then transmits a character, which, using standard protocols, includes three data bytes. If the bar code scanner **108** detects communications between the keyboard **110** and the PC **100**, the bar code scanner **108** waits until such communications are completed before inhibiting the keyboard **110** and transmitting the character. After transmission of the character, the bar code scanner **108** releases the inhibit. Then, the aforementioned cycle begins again until all decoded characters have been transmitted by the bar code scanner **108**.

This character-by-character inhibit was developed to allow for any additional communications that might be required between keyboard **110** and PC **100**. Certain country-specific keyboards as, for example, those utilized throughout Germany, require that status information for one or more keyboard LED indicators be sent from the PC **100** to the keyboard **110**, when the <shift> key is pressed. These keyboard LED indicators are also present on conventional keyboards used throughout the United States, and are used to indicate the status of the Num-Lock (numeric lock), Caps-Lock (capital lock), and Scroll-Lock keys. Illumination of an LED indicates that the corresponding "lock" function is active and operational for subsequent key presses, whereas non-illumination of an LED indicates that the corresponding function is inactive and not operational for subsequent key presses. The status of these LED indicators is tracked by the PC **100** and, therefore, the PC needs to communicate with the keyboard **110** so that the appropriate LED indicators will be illuminated or de-illuminated.

In operational environments where US-type keyboards are used, the character-by-character method allows for proper activation and deactivation of the LED indicators on the keyboard. For example, the caps-lock indicator will toggle properly. However, on many European keyboards, the caps-lock key functions differently than on U.S. keyboards. Pressing caps-lock a first time results in implementation of the caps-lock function on subsequent key presses. However, pressing caps-lock a second time implements a shifting function, instead of turning the caps-lock function off, as would be the case for a U.S. keyboard. The PC must accurately keep track of the number of times that the caps-lock key has been hit in order to provide proper drive signals to the keyboard LED indicators. However, if communications between the keyboard and the PC are subject to interruption, the PC may not be able to acquire accurate information about the correct operational status of the caps-lock, num-lock, and scroll-lock keys. Moreover, the PC may interpret received bar code characters as keyboard key presses, and attempt to update LED status information based upon one or more characters of a received bar code. If the character-by-character method is used in conjunction with a European keyboard, the PC may not acquire the necessary status information from the keyboard, or the PC may acquire superfluous information from scanned bar code symbols. The PC will send erroneous LED drive signals to the keyboard, causing inappropriate LEDs to be illuminated, and/or causing LEDs that should be illuminated to remain dark. Accordingly, every time that the bar code scanner sends a bar code to the keyboard controller port, it is then necessary to update the status of the LED indicators, or the indicators will not display accurate information.

Although the character-by-character method is useful in certain specific system applications, what would happen if keyboard **110** keys were being pressed at the same time that data was being scanned by the bar code scanner **108**? Bar code data and typed-in data would be intermingled, thereby corrupting both sets of data.

SUMMARY OF THE INVENTION

The intermingling of bar code data and typed-in data in a keyboard-wedge configuration is prevented through the use of a message-based keyboard inhibit procedure implemented by the bar code scanner. In this manner, both the typed-in data and the bar code data will remain uncorrupted, even if the keyboard data is being entered substantially simultaneously with the scanning and/or decoding of bar code data. This message-based keyboard inhibit procedure tests for any communication in progress between the keyboard and PC. If no communication is in progress, the bar code scanner places a switching mechanism into a first state so as to disable communications between the keyboard and a keyboard controller port of a PC (personal computer), thereby inhibiting the keyboard. The bar code scanner then transmits the decoded bar code to the PC as a sequence of data bytes. The bar code scanner also implements all communications that are required between the PC and the keyboard during this time. The inhibit will not be released until all characters of the bar code have been transmitted by the scanner. Typically, three data bytes are used to represent each bar code character, and bar codes include a plurality of such characters. Once all characters of the bar code are transmitted, the bar code scanner releases the inhibit by placing the switching mechanism into a second state, so as to permit communications to take place between the keyboard and the PC. At this time, any keyboard key that was typed in during transmission of the bar code will now be transmitted from the keyboard to the PC. Presently-existing keyboards have built-in buffers, and, therefore, such typed-in data will not be lost.

Pursuant to a further embodiment of the invention, the first state of the switching mechanism inhibits the keyboard by placing a keyboard clock line into a logic "low" state, and allowing a keyboard data line to float to a logic "high" state. The second state of the switching mechanism permits communication between the keyboard and the PC by allowing logic "high" pulses on the keyboard clock line.

Message-based keyboard inhibit solves the problem of the keyboard and bar code scanner communicating on the same channel without interference. The invention can be provided by equipping the bar code scanner with software, firmware, and/or operating instructions. The user of the bar code scanner will be able to type-in data simultaneously with the scanner scanning and/or decoding bar code data. Both sets of data will remain intact throughout the typing and scanning interval.

Pursuant to a further embodiment of the invention, the bar code scanner is equipped with an automatic detection mechanism to keep track of the status of one or more keyboard LED indicators, including at least one of the Caps-lock, Num-lock, and Scroll-lock LED indicators. The automatic detection mechanism is optionally equipped to keep track of a scan code transmission protocol currently in use, such as make-Break, Make-Only, AT or PS2-type scan-code transmission protocols. The manner in which the aforementioned LED indicators react to key presses varies according to the country type of the keyboard.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a hardware block diagram showing a prior art keyboard wedge bar code scanner configuration.

FIG. 2 is a hardware block diagram showing a keyboard wedge bar code scanner configuration embodying the techniques of the present invention.

FIGS. 3A–3K together comprise a flowchart setting forth a message-based keyboard inhibit procedure implemented by a bar code scanner according to a preferred embodiment of the invention.

FIGS. 4A–4C together comprise a flowchart setting forth a message-based keyboard inhibit procedure according to a preferred embodiment of the invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

FIG. 2 is a hardware block diagram showing a keyboard wedge bar code scanner configuration embodying the techniques of the present invention. Enhanced bar code scanner **208** is connected to a keyboard controller port **204** of a personal computer **200** via interconnection cable **206**. A first section of interconnection cable **206** connects keyboard controller port **204** to enhanced bar code scanner **208**, and a second section of interconnection cable connects enhanced bar code scanner **208** to computer keyboard **210**.

A digital switch **205**, controlled by enhanced bar code scanner **208**, is connected between the first and second sections of interconnection cable **206**. This digital switch **205** is used to selectively inhibit keyboard **210**. When the digital switch **205** is in a first state, the switch causes a keyboard clock line to remain in a logic "low" state, and allows a keyboard data line to float to a logic "high" state. In this state, the keyboard **210** is inhibited and cannot communicate with the keyboard controller port **204**. However, the enhanced bar code scanner **208** can now communicate with the keyboard controller port **204** without interference from keyboard **210**, even if keyboard keys are being pressed When the digital switch **205** is in a second state, the keyboard clock line is allowed to carry one or more clock pulses having a logic "high" state. In this manner, the keyboard **210** is enabled, and keyboard key presses will be transmitted to keyboard controller port **204**.

The manner in which the enhanced bar code scanner **208** selectively inhibits the keyboard **210** is described for illustrative purposes, as a switching mechanism may be employed to inhibit keyboard **210** using a technique other than, or in addition to, holding the clock line low. The scope of the invention includes such techniques. For example, the digital switch **205** could be placed in series between keyboard **210** and PC **200**. When in an open state, the switch **205** would cut off all communications between the keyboard **210** and the PC **200**, whereas, in a closed state, the switch **205** would connect the keyboard to the PC **200**.

Keyboard controller port **204** is driven by keyboard controller **202**. Interconnection cable **206** is shown for illustrative purposes only, as any other technique for conveying information from one place to another may be used in lieu of, or in addition to, interconnection cable **206**, as long as these communications are capable of being controlled by a switching mechanism that can selectively enable and disable the keyboard. These techniques may include wireless communication, wired communication, optical communication, and others. Accordingly, digital switch **205** is also shown for illustrative purposes, and another switching mechanism could be used in place of, or in addition to, digital switch **205**, depending upon the manner in which communications are to be provided between keyboard controller port **204** and keyboard **210**. Moreover, PC **200** is shown for illustrative purposes, as any of a variety of computing devices may be employed in lieu of, or in addition to, PC **200**. In the example of FIG. **2**, PC **200** includes a processor **201**. Enhanced bar code scanner **208** may contain an on-board processor **209**.

In the configuration of FIG. **2**, a computer keyboard **210** is employed to illustrate the potential problems that could occur if someone attempted to transmit a decoded bar code from the enhanced bar code scanner to **208** to the keyboard controller port **204** at the same time that the keyboard **210** is being used. If digital switch **205** is in the aforementioned second state, bar code data can interfere with keyboard data. However, if digital switch **205** is in the aforementioned first state, the computer is unable to send LED status indication signals to the keyboard, and, moreover, data typed into the keyboard will not reach the keyboard controller port.

Enhanced bar code scanner **208** is "enhanced" in the sense that it is equipped to execute a procedure for controlling a switching mechanism such as, for example, digital switch **205**, so as to selectively disable keyboard **210**. The enhancement may be implemented using software, firmware, and/or operating instructions. Use of the term "enhanced" is not necessarily indicative of any hardware enhancements to the bar code scanner.

The enhanced bar code scanner **208** controls the switching mechanism so as to permit substantially simultaneous operation of computer keyboard **210** and enhanced bar code scanner **208**. The intermingling of bar code data and typed-in data in the keyboard-wedge configuration of FIG. **2** is prevented through the use of a message-based keyboard inhibit procedure implemented by the enhanced bar code scanner **208**. In this manner, both the typed-in data and the bar code data will remain uncorrupted, even if the keyboard data is being entered substantially simultaneously with the scanning and/or decoding of bar code data.

Refer now to FIGS. 4A–4C, which together comprise a flowchart setting forth a message-based keyboard inhibit procedure according to a preferred embodiment of the invention. The procedure commences at block **401** where the enhanced bar code scanner tests for any communication in progress between keyboard **210** and PC **200** (FIG. 2). If communications are in progress, the bar code scanner waits for a predetermined amount of time at block **403**, and the program then loops back to block **401**. If no communication is in progress, the negative branch from block **401** to block **405** is followed. The enhanced bar code scanner **208** (FIG. 2) places a switching mechanism (such as, for example, digital switch **205**) into the first state so as to disable communications between keyboard **210** and keyboard controller port **204**, thereby inhibiting the keyboard.

Next, at block **407**, the enhanced bar code scanner **208** transmits the decoded bar code to the PC **200** as a sequence of data bytes. The enhanced bar code scanner **208** implements any communications that may be required between the PC and the keyboard during this time, emulates keyboard responses to PC interrogations; and/or stores information sent by the PC and related to the status of one or more visual indicators on the keyboard (block **409**). This stored information is used to ascertain the actual status of these one or more visual indicators, such that the bar code scanner causes accurate visual indicator drive signals to be sent to the keyboard when the keyboard inhibit is released. These visual indicators may indicate the status of one or more keyboard **210** functions, such as caps-lock, num-lock, and scroll-lock. The inhibit will not be released (block **413**) until all characters of the bar code have been transmitted by the scanner (block **411**). Typically, three data bytes are used to represent each bar code character, and bar codes include a plurality of such characters. Once all characters of the bar code are transmitted, the enhanced bar code scanner releases the inhibit (block **413**) by placing the switching mechanism (i.e., digital switch **205**) into the second state. At this time (block **415**), any keyboard key that was typed in during transmission of the bar code will now be transmitted from the keyboard **210** to the PC **200**. Presently-existing keyboards have built-in buffers, and, therefore, such typed-in data will not be lost.

Message-based keyboard inhibit solves the problem of the keyboard and bar code scanner communicating on the same channel without interference. The invention can be provided by equipping the bar code scanner with software, firmware, and/or operating instructions. The user of the bar code scanner will be able to type-in data simultaneously with the scanner scanning and/or decoding bar code data. Both sets of data will remain intact throughout the typing and scanning interval.

Pursuant to a further embodiment of the invention, while the computer keyboard **210** is inhibited by digital switch **205**, the enhanced bar code scanner **208** automatically detects the actual status of one or more indicators on computer keyboard **210**. These indicators may be provided in the form of one or more LED lamps indicative of the status of the Caps-lock, Num-lock, and Scroll-lock keys. The procedure for automatically detecting LED indicator status is as follows. The enhanced bar code scanner is programmed to send to the PC data bytes representing two "num-lock" key toggles. The PC responds to each of the "num-lock" data bytes by sending keyboard LED status indicator drive signals to the keyboard controller port. The enhanced bar code scanner intercepts the drive signals for each occurrence of "num-lock" to thereby ascertain the actual status of the caps-lock key.

The enhanced bar code scanner **208** optionally keeps track of the transmission protocol currently in use, such as make-Break,Make-Only, AT or PS2-type scan-code transmission protocols. These status indications and transmission protocols may vary, depending upon the keyboard country type selected, and are maintained by the enhanced bar code scanner **208** until communications between keyboard **210** and PC **200** are resumed.

Pursuant to a still further embodiment of the invention, the enhanced bar code scanner **208** is equipped to detect a keyboard inhibit signal at the keyboard controller port **204**. This inhibit signal may be generated by PC **200**), and/or by any of a number of devices coupled to, and/or within this computing device. For example, the inhibit signal may be generated by the PC **200** pulling the clock line to a logic "low" state. In this example, the enhanced bar code scanner **208** would detect the existence of a keyboard inhibit signal by monitoring the clock line.

If the enhanced bar code scanner **208** detects a keyboard inhibit signal while a data byte is being transmitted to the keyboard controller port **204**, the bar code scanner retransmits this data byte to the keyboard controller port **204**. This retransmission process is repeated up to a specified number of times, so as to provide additional opportunities for the data byte to be inputted to the keyboard controller port **204** if the port is momentarily disabled by the keyboard inhibit signal. For many applications, it is advantageous to repeat the retransmission process up to three times for a given data byte. If the data byte is still not successfully received after the third attempt, the process is no longer repeated. This data byte may represent one or more scan codes which, in turn, represent one or more keyboard key presses or decoded bar codes. One purpose of this retransmission procedure is to accurately transmit a bar code on a transmission line that is subjected to possible halt or inhibit signals issued by PC **200**. It should be noted that any number of sources contained within PC **200** could generate these inhibit signals. When a data byte is received at the keyboard port, this byte, which may represent all or part of a decoded bar code, is optionally displayed on a monitor coupled to PC **200**.

Refer to FIGS. 3A–3K which together comprise a flowchart setting forth a message-based keyboard inhibit procedure implemented by a bar code scanner according to a preferred embodiment of the invention. In overview, the method is organized into a first sequence of steps related to power-up and diagnostics (blocks **301–304**), a second sequence of steps related to the decoding and storing of data (blocks **305–306**), a third sequence of steps related to keyboard output (blocks **307–308**), a fourth sequence of steps related to ASCII to scan code conversion (blocks **309–311**), a fifth sequence of steps related to preparation for a keyboard output sequence (blocks **312–317**), a sixth sequence of steps related to first character keyboard initialization (blocks **318–322**), a seventh sequence of steps related to keyboard scan code output (blocks **323–327**), an eighth sequence of steps related to preparation for exiting the keyboard scan code transmission program (blocks **328–333**), and a ninth sequence of steps related to a determination of whether or not there are more characters to transmit (blocks **334–335**). It should be noted that the above-described organization of the steps into nine sequences is set forth for purposes of illustration and convenience. The steps may, but need not, be organized in the manner described above.

First Sequence of Steps

Power-up and Power on Diagnostics

The program of FIGS. 3A–3K commences at block **301**, where power is applied to enhanced bar code scanner (**208**, FIG. **2**) and/or PC (**200**, FIG. **2**) Next, at block **302** (FIG. **3**), the program determines the operational mode of the enhanced bar code scanner. The operational mode is "normal" if the enhanced bar code scanner is to be used in a Y or wedge configuration with a computer keyboard, similar to the configuration of FIG. **1**. The mode is "stand-alone" if, as shown in FIG. **2**, the enhanced bar code scanner is a "stand alone" unit not to be used with a separate computer keyboard. If the operational mode is "normal", the program skips ahead to block **305**. Otherwise, the program proceeds to block **303**.

At block **303**, it has already been determined that a separate computer keyboard (refer to FIG. **1**) is not present. Accordingly, the enhanced bar code scanner (FIG. **2**) must communicate to PC **200** all command bytes for successful completion of power-on diagnostics. The enhanced bar code scanner begins this keyboard emulation process by transmitting an "AA" to indicate that the "keyboard" self-test was completed successfully. After this point, the PC and "keyboard" (as emulated by the enhanced bar code scanner) may communicate and acknowledge any of a number of various commands to complete the power-on diagnostic sequence. Some PC's require more information than others for the power-on diagnostic procedure. Therefore the number of commands sent by the PC and acknowledged by the emulated keyboard is variable. These commands may include any of the following: Typamatic Rate, Make/Break status, scan code set enabled, and LED status.

At block **304**, once the sequence of command bytes of block **303** are completed, the PC will power up normally. Note that powering up is sometimes referred to as "booting up". Without successful completion of these commands, the PC will fail to power up. After the PC boots up, the enhanced bar code scanner is ready to decode data.

Second Sequence of Steps

Decoding and Storing Data

At block **305**, bar code symbols are decoded into human-readable and/or ASCII characters which are then stored in a memory device associated with the enhanced bar code scanner, such as, for example, random-access memory (RAM). At block **306**, the program generates an appropriate output interface equipped to select output data from the enhanced bar code scanner.

Third Sequence of Steps

Begin Keyboard Output

A test is performed at block **307** to determine the output mode of the emulated keyboard. Typical output modes include AT, XT, and PS2. At block **308**, the program determines the type of keyboard transmission that is to take place. Types of keyboard transmission include normal, ALT Mode, and special key transmission sequences.

Fourth Sequence of Steps

ASCII to Scan Code Conversation

At block **309**, if the type of keyboard transmission (as determined at block **308**) is normal transmission, the program proceeds to blocks **310** & **311**. Otherwise, the program jumps ahead to block **312**. The country output setting of the keyboard is determined at block **310**. This country output setting specifies any of a plurality of keyboards, such as US (United States), UK (United Kingdom), FR (France), GR (Germany), BE (Belgium), SP (Spain), IT (Italy), and JP (Japan). Next, at block **311**, an ASCII character of a bar code string received by the enhanced bar code scanner is converted to an equivalent scan code or keyboard key code for the particular keyboard type or country type determined at block **310**.

Fifth Sequence of Steps

Prepare for Keyboard Output Sequence

The program proceeds (block **312**) with the outputting of scan codes based on the keyboard type as previously determined at block **310**. Then, at block **313**, a test is performed to ascertain whether a "stand alone" (FIG. **2**) or "normal keyboard wedge" configuration is in use. The "normal keyboard wedge" configuration is similar to FIG. **1**, except that an enhanced bar code scanner is used in place of bar code scanner **108**. At block **314**, program control is routed to an appropriate operational sequence, depending upon the configuration in use as determined at block **313**. If a "normal keyboard wedge" configuration is in use, the program proceeds with blocks **315**, **316**, and **317**. Otherwise, the program jumps ahead to block **318**.

At block **315**, a test is performed to ascertain whether message-based inhibit or character inhibit has been selected. If message inhibit has been selected, then the program skips ahead to block **318**. Otherwise, the program proceeds to block **316**. A test is performed at block **316** to check for any PC to keyboard communications currently taking place. If there are any currently active communications taking place, the program waits until any such communication has ended. Then (block **317**), the keyboard is inhibited from the PC.

Sixth Sequence of Steps

First Character Keyboard Initialization

This sequence commences by performing a test to determine whether or not a 1st character of a scanner bar code transmission sequence has been sent by the enhanced bar code scanner (block **318**). This scanner bar code transmission sequence is organized into a data byte, as previously described. If not, the program jumps ahead to block **323**. Otherwise, the program proceeds to block **319**, where a check is performed to determine whether or not normal keyboard transmission is required. If so, the program proceeds to block **320**. If not, the program jumps ahead to block **323**.

Next, a test is performed to determine if a European keyboard type (FR, GR, BE).is in use. If so, the program continues to block **321**. If not, the program jumps ahead to block **323**. At block **321**, a [SHIFT] is transmitted and the PC releases the keyboard inhibit so as to place the keyboard into a known (lower-case) operational state. The PC then re-enables the keyboard inhibit (block **322**).

Seventh Sequence of Steps

Keyboard Scan Code(s) are Outputted

At this point (block **323**), if a normal keyboard wedge scanner configuration is in use, the scanner will communicate with other system components as if it was a stand-alone bar code scanner. All communications, if any, that are required will be acknowledged by the bar code scanner. Next (block **324**), the output sequence is determined. The output sequence specifies any of Normal, Alt Mode, and Special Key transmission. All of these modes require the transmission of keyboard scan codes. Alt mode transmits the decimal value of the ASCII value using the numeric keypad scan codes. Special Key transmissions are the transmission of non-ASCII keys (i.e. function keys—F1, F2, etc., arrow keys and so forth). Normal key transmission is the transmission of normal ASCII characters that have an associated

keyboard scan code. All scan codes may be transmitted using the aforementioned "3 strikes technique" to output the data.

The enhanced bar code scanner implements a keyboard scan code transmission (block **325**) using a plurality of data bytes. A "MAKE" scan code followed by a 2 byte scan code BREAK sequence is transmitted. All **3** scan codes can use the "3 strikes" method. At block **326**, each keyboard scan code transmission is an 11-bit output sequence. Its format is as follows: Start bit (**0**), 8 data bits (scan code), parity bit (odd parity) and a stop bit (**1**). While transmitting the bits, the PC can halt the transmission of the scan code by pulling the keyboard clock or data line low. Depending on the state of these lines, the keyboard may need to respond to the PC or wait until the lines go back to a ready-state and retransmit the scan code, up to a predetermined maximum number of times. This predetermined number is three in cases where the "3 strikes technique" is employed.

Once the MAKE-BREAK sequence is complete (block **327**), then an intercharacter delay is implemented.

Eighth Sequence of Steps

Prepare to Exit keyboard Scan

Code Transmission Routine

A test is again performed to ascertain whether a "stand alone" (FIG. **2**) or "normal keyboard wedge" configuration is in use (block **328**). If a Stand alone keyboard scanner is in use, the program jumps ahead to block **334**. Otherwise, the program continues to block **329**, where the program checks for a Message Inhibit and a Character Inhibit. If a Character Inhibit exists, this inhibit is released, and the program jumps ahead to block **334**. If a Message Inhibit exists, the program advances to block **330**, where a check is performed to determine whether or not this is the last character of the message. If it is the last character, the program advances to block **331** whereas, if it is not., the program advances to block **334**.

At block **331**, since this is the last character of the message, it is time to release the inhibit. It must also be determined if a scan code is to be transmitted that could have had an effect on the keyboard LED status. Check if any of the following scans codes are to be transmitted: Caps: Lock, Num Lock, Right or Left Shift. If not, continue at Block **334**. Next (block **332**), scan codes of Right or Left Shift only have keyboard LED status indicator implications with European keyboards. Accordingly, a check is performed to determine whether or not a FR, GR, or BE keyboard is in use. If any of these keyboards is in use, transmit the Shift scan code and allow the keyboard and PC to communicate; otherwise continue to block **333**. If a scan code of Num Lock or Caps Lock is to be transmitted, transmit the appropriate scan code and allow the keyboard and PC to communicate, but correct the keyboard LED status indicators.

Ninth Sequence of steps

Determine Last Character to be Transmitted

Check if all characters have been transmitted (block **334**). If not, the next ASCII character is obtained, the corresponding scan code is transmitted, and the program loops back to block **307**. At block **335**, the keyboard scan code output has been completed.

The program of FIGS. **3A–3K** has been organized into nine operational sequences for illustrative purposes and as an explanatory aid. Those skilled in the art can appreciate that the program need not necessarily be organized into nine operational sequences, as variations thereto are still within the spirit and scope of the invention.

As stated above, the keyboard emulation mechanism may include software, firmware, and/or operating instructions. Use of this term herein is not necessarily indicative of any hardware enhancements to the bar code scanner. If the scanner is programmed to allow for up to three retransmissions, such a method is referred to as the "3-strikes-technique". A conventional transmission of one pressed key consists of three (3) characters—a make scan code and two (2) break code characters. Therefore, this technique could possibly transmit up to 9 characters to complete the key sequence. This retransmit technique gives the enhanced bar code scanner a limited number of retries per scan code, thus allowing a transmission sequence to move forward instead of hanging or sticking while trying to retransmit the same character over and over again.

But why should there ever be a need to retransmit a scan code? The retransmit protocol was developed after noticing that other devices or software within the PC could also inhibit the keyboard controller port. Some devices/software such as network drivers may inhibit the keyboard controller port every few milliseconds to perform some function without interrupt. Other devices, such as a mouse, may be on the same interrupt as the keyboard controller port, thus requiring the keyboard controller port to be inhibited while the software governing the mouse moves the mouse cursor. With laptops, the same types of inhibit issues can exist. However, even more restraints may be placed on the external keyboard controller port of a laptop than would be the case with a conventional PC. This renders use of the keyboard controller port as an auxiliary input channel all the more difficult in the operational environment of laptop computing.

With the advent of Windows 3.1 and Window 95, it appeared that more and more devices/software were effecting the behavior of the keyboard controller port. Therefore, it was imperative that a technique was developed that would transmit the scan code accurately, but take into account the fact that during transmission the keyboard could receive an inhibit and a retransmit would be necessary to accurately transmit the decoded bar code. The "3-strikes-technique" set forth herein accomplishes this task uniquely within a limited transmission sequence.

We claim:

1. A method for use with a bar code scanner that is adapted for connection to: (i) a keyboard controller port of a computing device, and (ii) a computer keyboard, in a keyboard-wedge configuration, the method preventing an intermingling of bar code data acquired by the bar code scanner and typed-in keyboard data typed into the keyboard by performing the steps of:

(a) the bar code scanner testing for the existence of any communication in progress between the keyboard and the computing device;

(b) if no such communication is in progress, the bar code scanner causing a switching mechanism to enter a first state so as to disable communications between the computer keyboard and the keyboard controller port, thereby inhibiting the keyboard;

(c) the bar code scanner transmitting a decoded bar code to the keyboard controller port as a sequence of data bytes, and/or emulating any communications that may be required between the keyboard controller port and the keyboard;

(d) once the bar code scanner has transmitted all characters of the bar code to the keyboard controller port, the bar code scanner releasing the keyboard inhibit by causing the switching mechanism to enter a second state, so as to permit communications between the keyboard and the keyboard controller port.

2. The method of claim **1** wherein step (d) further includes the step of the keyboard transmitting to the keyboard con-

troller port any keyboard key that was typed in during transmission of the bar code in step (c).

3. The method of claim 1 wherein the first state of the switching mechanism inhibits the keyboard by placing a keyboard clock line into a logic "low" state, and allowing a keyboard data line to float to a logic "high" state.

4. The method of claim 3 wherein the second state of the switching mechanism permits communication between the keyboard and the PC by allowing logic "high" pulses on the keyboard clock line.

5. The method of claim 1 wherein the computing device is a PC (personal computer) or a laptop computer.

6. The method of claim 1 wherein the bar code scanner performs steps (a)–(d) using software, firmware, and/or operating instructions.

7. The method of claim 1 further including the steps of: a user of the bar code scanner typing on the keyboard substantially simultaneously with the user scanning and/or decoding bar code data; and the keyboard data remaining separate from the bar code data while the typing, scanning, and/or decoding is being performed.

8. The method of claim 1 for use with a computer keyboard having one or more visual indicators indicating a status of a corresponding keyboard function including at least one of capital lock (caps-lock), numeric lock (num-lock), and scroll lock (scroll-lock), wherein the corresponding keyboard function has an enabled status and a disabled status, the method further including the step of:

the bar code scanner keeping track of the status of the one or more visual indicators while communications between the computer keyboard and the keyboard controller port is disabled in step (b).

9. The method of claim 8 further including the step of the bar code scanner automatically determining the status of the corresponding keyboard function by transmitting bytes representative of a first and a second emulated num-lock key press to the keyboard and monitoring keyboard responses to each of the first and second emulated num-lock key presses to thereby determine the status of the corresponding keyboard function.

10. The method of claim 8 further including the step of the bar code scanner keeping track of a scan code transmission protocol currently in use, wherein the scan code transmission protocol includes at least one of Make-Break, Make-Only, AT, and PS2.

11. The method of claim 8 further including the steps of:

the bar code scanner interrogating the keyboard; and

the keyboard, in response to the interrogating, transmitting to the keyboard an indicia of a keyboard country type uniquely identifying a country with which the keyboard is associated.

12. A bar code scanner adapted for connection to: (i) a keyboard controller port of a computing device, and (ii) a computer keyboard, in a keyboard-wedge configuration, the bar code scanner having a keyboard inhibit mechanism for preventing an intermingling of bar code data acquired by the bar code scanner and typed-in keyboard data typed into the keyboard; the keyboard inhibit mechanism comprising:

(a) a monitoring mechanism adapted to test for the existence of any communication in progress between the keyboard and the computing device;

(b) a switching mechanism, coupled to the monitoring mechanism, wherein, in response to an absence of communication in progress between the keyboard and the computing device, the switching mechanism enters a first state so as to disable subsequent communications between the computer keyboard and the keyboard controller port, thereby inhibiting the keyboard;

(c) a bar code transmission mechanism, responsive to the switching mechanism being in the first state, for transmitting a decoded bar code to the keyboard controller port as a sequence of data bytes, and/or for emulating any communications that may be required between the keyboard controller port and the keyboard;

wherein the switching mechanism is responsive to the code transmission mechanism such that, once the bar code transmission mechanism has transmitted all characters of the bar code to the keyboard controller port, the switching mechanism enters a second state so as to release the keyboard inhibit and to permit communications between the keyboard and the keyboard controller port.

13. The bar code scanner of claim 12 wherein the keyboard includes a buffering mechanism such that the keyboard is adapted to transmit to the keyboard controller port one or more keyboard keys that were typed in during transmission of the bar code to the keyboard controller port.

14. The bar code scanner of claim 12 wherein the first state of the switching mechanism inhibits the keyboard by placing a keyboard clock line into a logic "low" state, and allowing a keyboard data line to float to a logic "high" state;

and wherein the second state of the switching mechanism permits communication between the keyboard and the PC by allowing logic "high" pulses on the keyboard clock line.

15. The bar code scanner of claim 12 wherein the computing device is a PC (personal computer) or a laptop computer.

16. The bar code scanner of claim 12 wherein the monitoring mechanism, switching mechanism, and bar code transmission mechanism are implemented using software, firmware, and/or operating instructions.

17. The bar code scanner of claim 12 wherein, in response to a user of the bar code scanner typing on the keyboard substantially simultaneously with the user scanning and/or decoding bar code data, the keyboard data remains separate from the bar code data while the typing, scanning, and/or decoding is being performed.

18. The bar code scanner of claim 12 for use with a computer keyboard having one or more visual indicators indicating a status of a corresponding keyboard function including at least one of capital lock (caps-lock), numeric lock (num-lock), and scroll lock (scroll-lock), wherein the corresponding keyboard function has an enabled status and a disabled status, the bar code scanner further including an automatic detection mechanism for keeping track of the status of the one or more visual indicators while communications between the computer keyboard and the keyboard controller port is disabled.

19. The bar code scanner of claim 18 wherein the automatic detection mechanism automatically determines the status of the corresponding keyboard function by transmitting bytes representative of a first and a second emulated num-lock key press to the keyboard and monitoring keyboard responses to each of the first and second emulated num-lock key presses to thereby determine the status of the corresponding keyboard function.

20. The bar code scanner of claim 18 further including a protocol detection mechanism by which the bar code scanner keeps track of a scan code transmission protocol currently in use, the scan code transmission protocol including at least one of Make-Break, Make-Only, AT, and PS2.

* * * * *

US005875415A

United States Patent [19]

Lieb et al.

[11] Patent Number: 5,875,415

[45] Date of Patent: Feb. 23, 1999

[54] **UNIVERSAL HOST INTERFACE FOR DATA ACQUISITION SYSTEMS**

[75] Inventors: **Robert Lieb**, Islip; **Nicholas Buongervino**, Hauppauge; **Patrick Mauro**, Lake Grove; **Michael Rizzi**, N. Babylon; **Ellen Oppenheim**, Coram; **Joseph Boriotti**, E. Northport; **Robert May**, Sayville, all of N.Y.; **Sara Karcher**, Boise, Id.; **Anthony Biuso**, S. Setauket, N.Y.; **Paul R. Poloniewicz**, E. Setauket, N.Y.; **Daniel Brown**, E. Northport, N.Y.; **Anthony Fama**, Mastic, N.Y.; **Robert Stein**, Plainview, N.Y.; **Cary Chu**, Port Jefferson, N.Y.; **Michael Catalano**, Lindenhurst, N.Y.; **Altaf Mulla**, Merrick, N.Y.

[73] Assignee: **Symbol Technologies, Inc.**, Holtsville, N.Y.

[21] Appl. No.: **932,496**

[22] Filed: **Sep. 18, 1997**

Related U.S. Application Data

[63] Continuation of Ser. No. 439,833, May 12, 1995, abandoned.

[51] **Int. Cl.**^{6} **H03K 13/24**

[52] **U.S. Cl.** **702/122**; 702/91; 395/883

[58] **Field of Search** .... 235/375, 462, 235/463; 364/932.62, 238.3, 239, 514 R, 514 C; 395/836, 883; 702/90, 91, 122

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,500,933	2/1985	Chan	364/950
5,258,604	11/1993	Behrens et al.	235/462
5,548,510	8/1996	Ebert et al.	364/443

OTHER PUBLICATIONS

Instrumentation for Engineering Measurements Dally et al. John Wiley & Sons, Inc Publisher, pp. 99–104, 1993.

MicroSoft Press Computer Dictionary 2nd Edition Woodcock et al. Publisher: Microsoft Press, 1993.

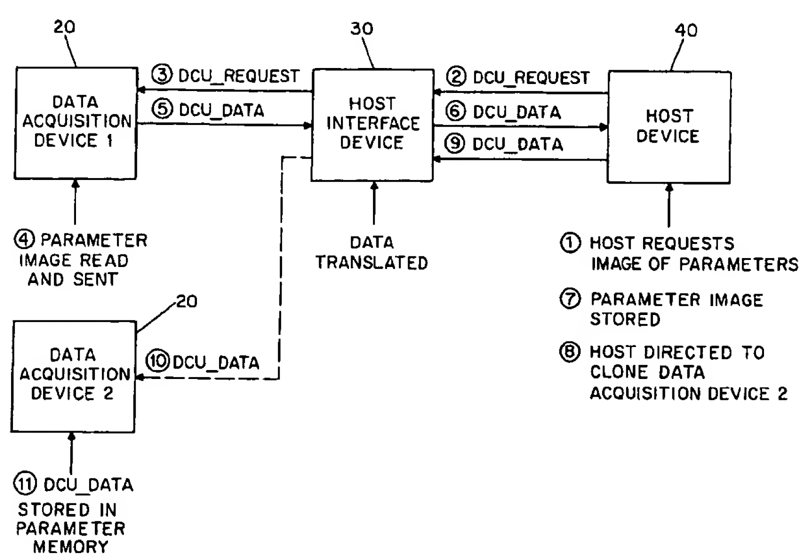
Donald Watts, I. Davies (Davies et al.) Computer Network and Their Protocols, 1979, pf 4, John Wiley & Sons Ltd.

Primary Examiner—John Barlow
Assistant Examiner—Craig Steven Miller

[57] **ABSTRACT**

A universal interface system for use with a data acquisition system including a data acquisition device such as a bar code scanner or scanner integrated terminal for providing digital data signals indicative of acquired data for transmission to a host device such as a computer, and a host device having host-specific input/output data format requirements. The universal interface system comprises universal data exchange means located in the data acquisition device for providing digital data signals in a universal data exchange format independent of the data format requirements of the host device; and a host interface cable coupled to the data acquisition device and to said host device, for transmitting acquired digital data from the data acquisition device to the host device, the interface cable comprising means for translating digital data from the universal data exchange format to the host-specific input/output data format requirements. The interface cable comprises a buffer for storing a host parameter data word used in communications with the host device. The universal data exchange means comprises a memory for storing a plurality of host parameter data words, each of the host parameter data words being associated with one of a plurality of different host interface modules; and means for reading from the memory and transmitting to the host interface module the host parameter data word associated with the particular host interface module operatively connected thereto.

23 Claims, 7 Drawing Sheets



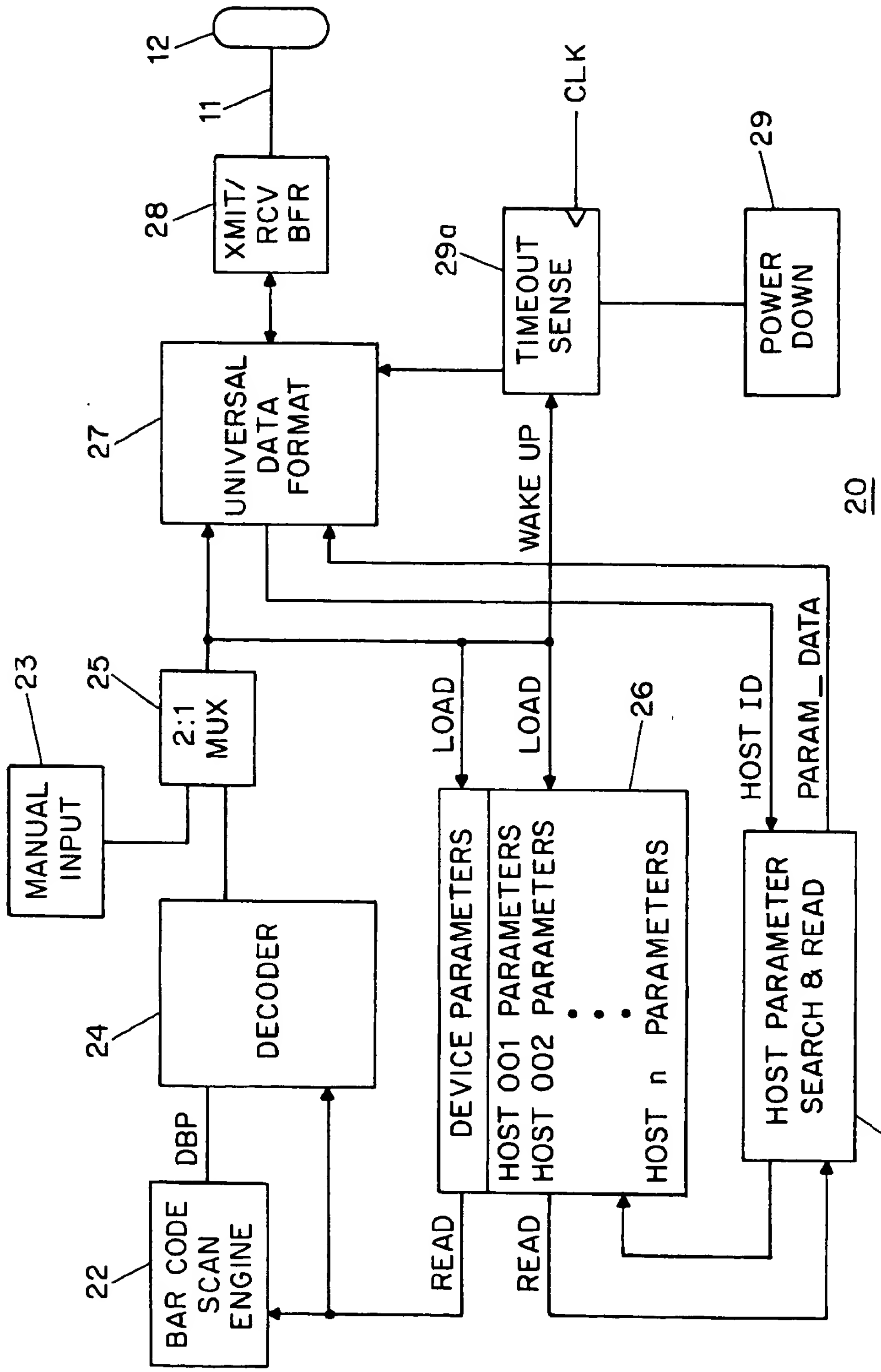


FIG. 1(A)

10/29/2003, EAST Version: 1.4.1

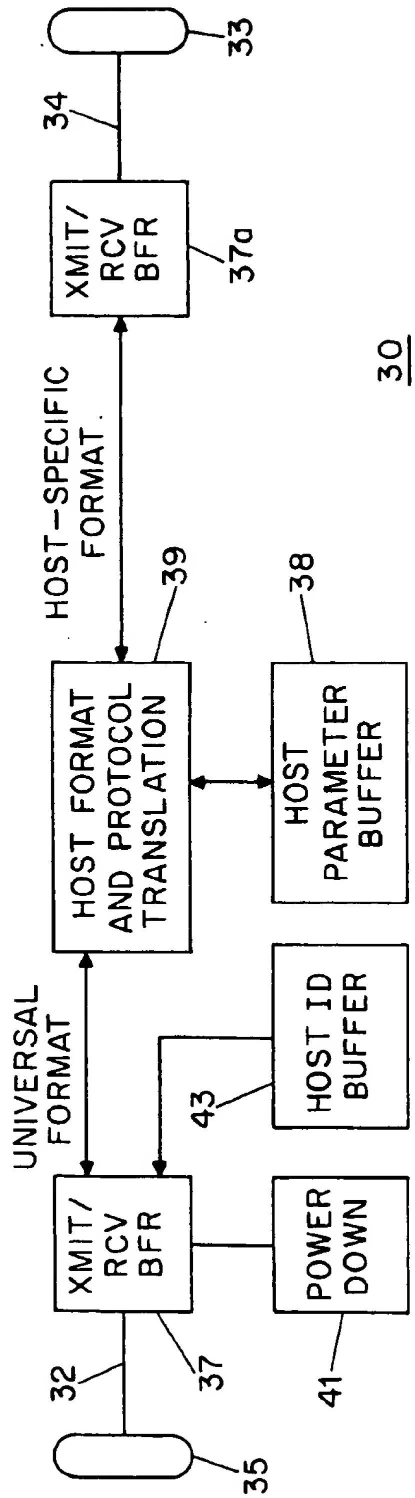

XMIT/ RCV BFR
HOST FORMAT AND PROTOCOL TRANSLATION
XMIT/ RCV BFR
UNIVERSAL FORMAT
HOST-SPECIFIC FORMAT
HOST PARAMETER BUFFER
HOST ID BUFFER
POWER DOWN
30
32
33
34
35
37
37a
38
39
41
43

FIG. 1(B)

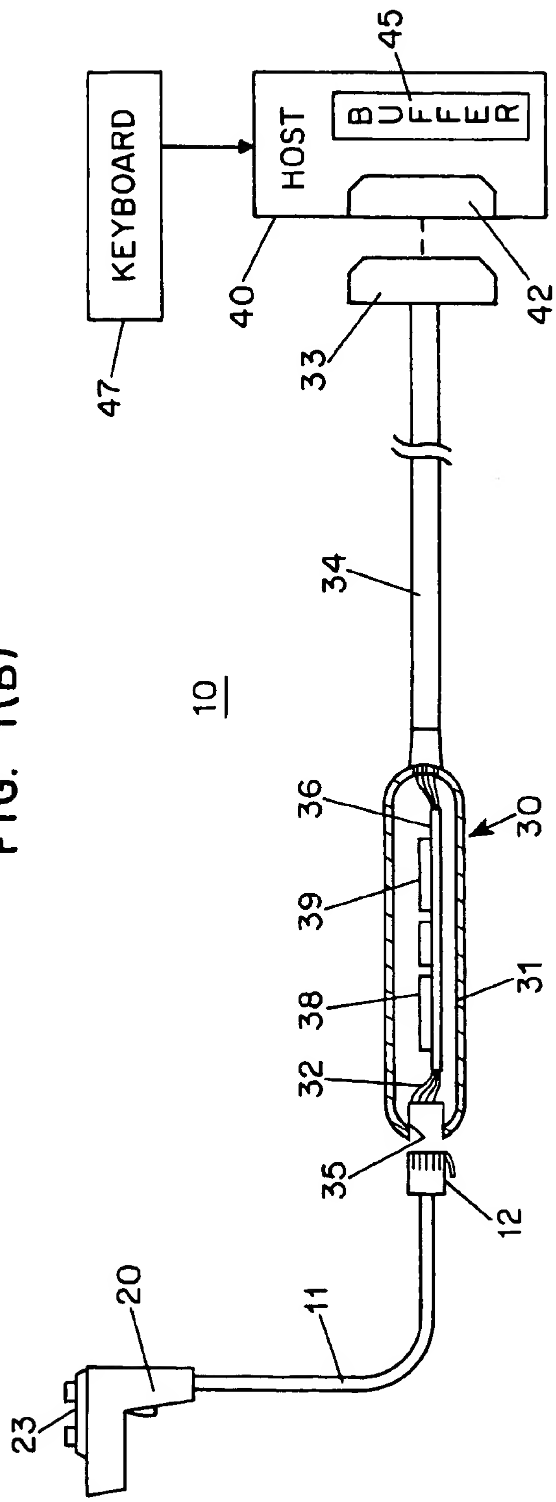

KEYBOARD
HOST
BUFFER
10
11
12
20
23
30
31
32
33
34
35
36
38
39
40
42
45
47

FIG. 2

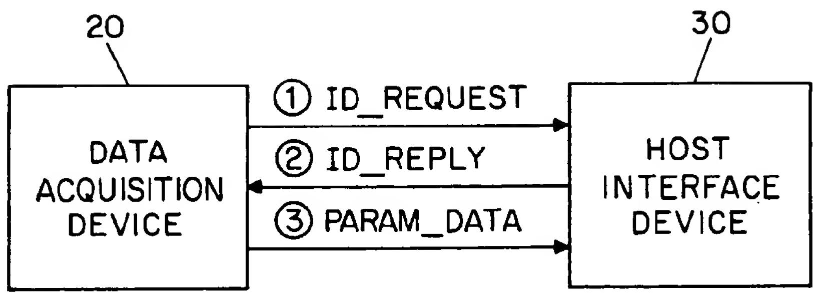

20
30
DATA ACQUISITION DEVICE
1 ID_REQUEST
2 ID_REPLY
3 PARAM_DATA
HOST INTERFACE DEVICE

FIG. 3(A)

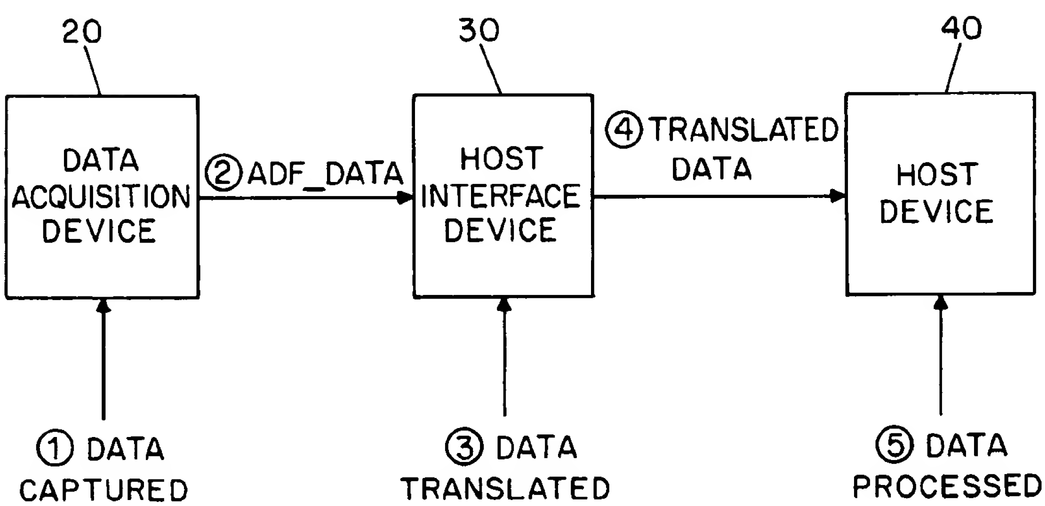

20
30
40
DATA ACQUISITION DEVICE
2 ADF_DATA
HOST INTERFACE DEVICE
4 TRANSLATED DATA
HOST DEVICE
1 DATA CAPTURED
3 DATA TRANSLATED
5 DATA PROCESSED

FIG. 3(B)

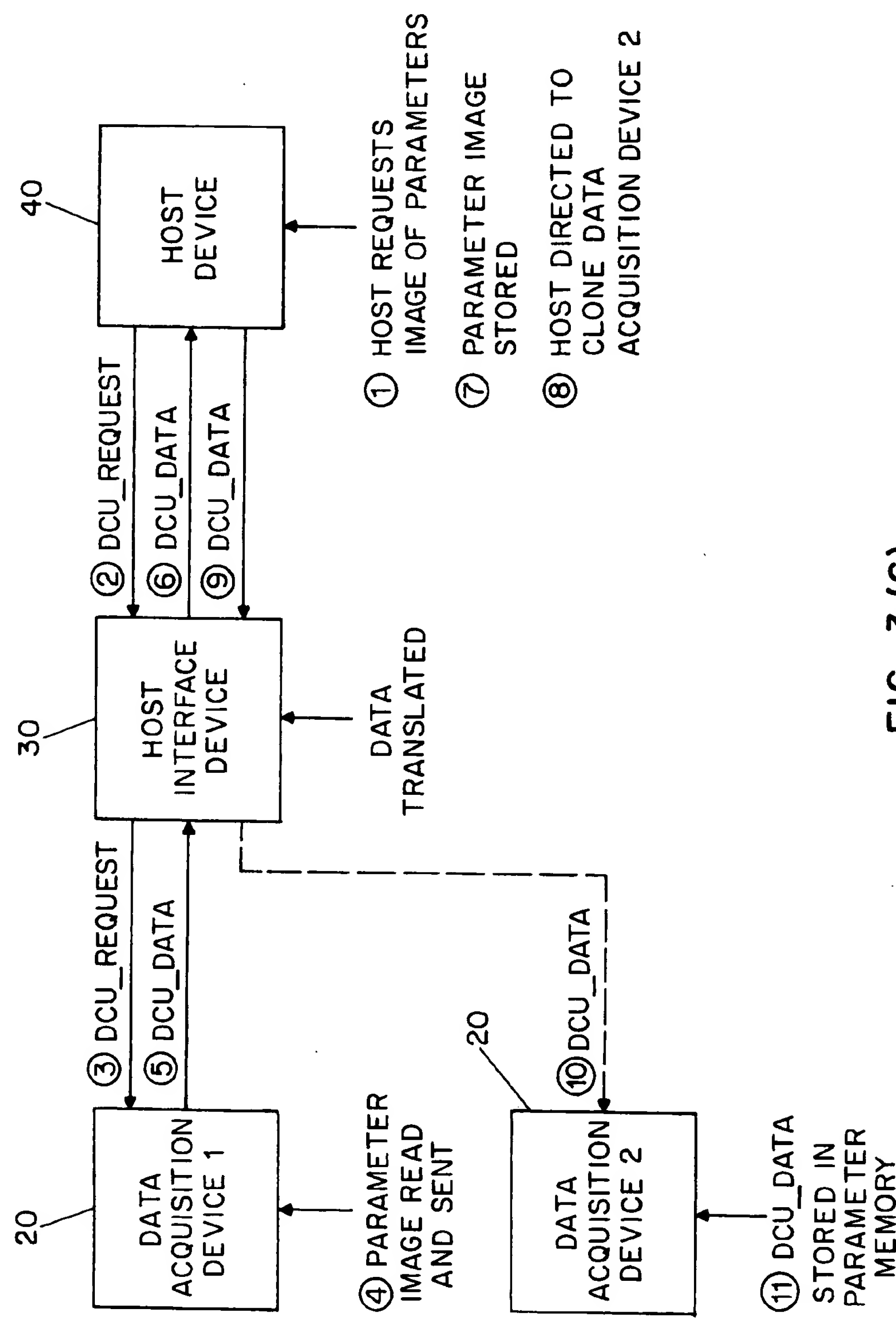

40
HOST DEVICE
30
HOST INTERFACE DEVICE
20
DATA ACQUISITION DEVICE 1
20
DATA ACQUISITION DEVICE 2
① HOST REQUESTS IMAGE OF PARAMETERS
② DCU_REQUEST
③ DCU_REQUEST
④ PARAMETER IMAGE READ AND SENT
⑤ DCU_DATA
⑥ DCU_DATA
DATA TRANSLATED
⑦ PARAMETER IMAGE STORED
⑧ HOST DIRECTED TO CLONE DATA ACQUISITION DEVICE 2
⑨ DCU_DATA
⑩ DCU_DATA
⑪ DCU_DATA STORED IN PARAMETER MEMORY

FIG. 3(C)

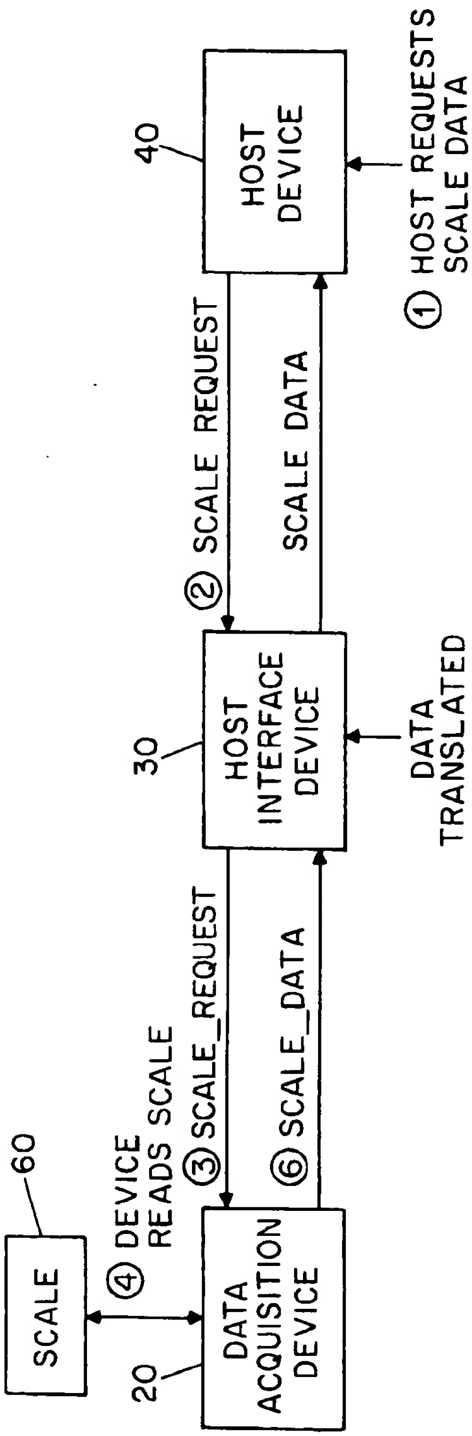

SCALE
60
DATA ACQUISITION DEVICE
20
HOST INTERFACE DEVICE
30
HOST DEVICE
40
① HOST REQUESTS SCALE DATA
② SCALE REQUEST
SCALE DATA
DATA TRANSLATED
③ SCALE_REQUEST
④ DEVICE READS SCALE
⑥ SCALE_DATA

FIG. 3(D)

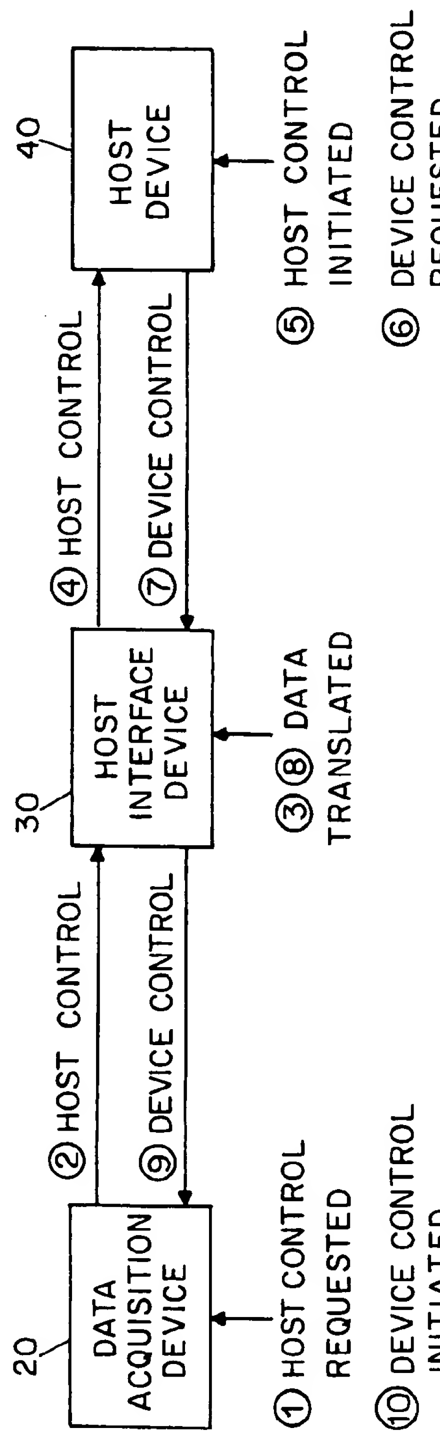

DATA ACQUISITION DEVICE
20
HOST INTERFACE DEVICE
30
HOST DEVICE
40
① HOST CONTROL REQUESTED
⑩ DEVICE CONTROL INITIATED
② HOST CONTROL
⑨ DEVICE CONTROL
③ ⑧ DATA TRANSLATED
④ HOST CONTROL
⑦ DEVICE CONTROL
⑤ HOST CONTROL INITIATED
⑥ DEVICE CONTROL REQUESTED

FIG. 3(E)

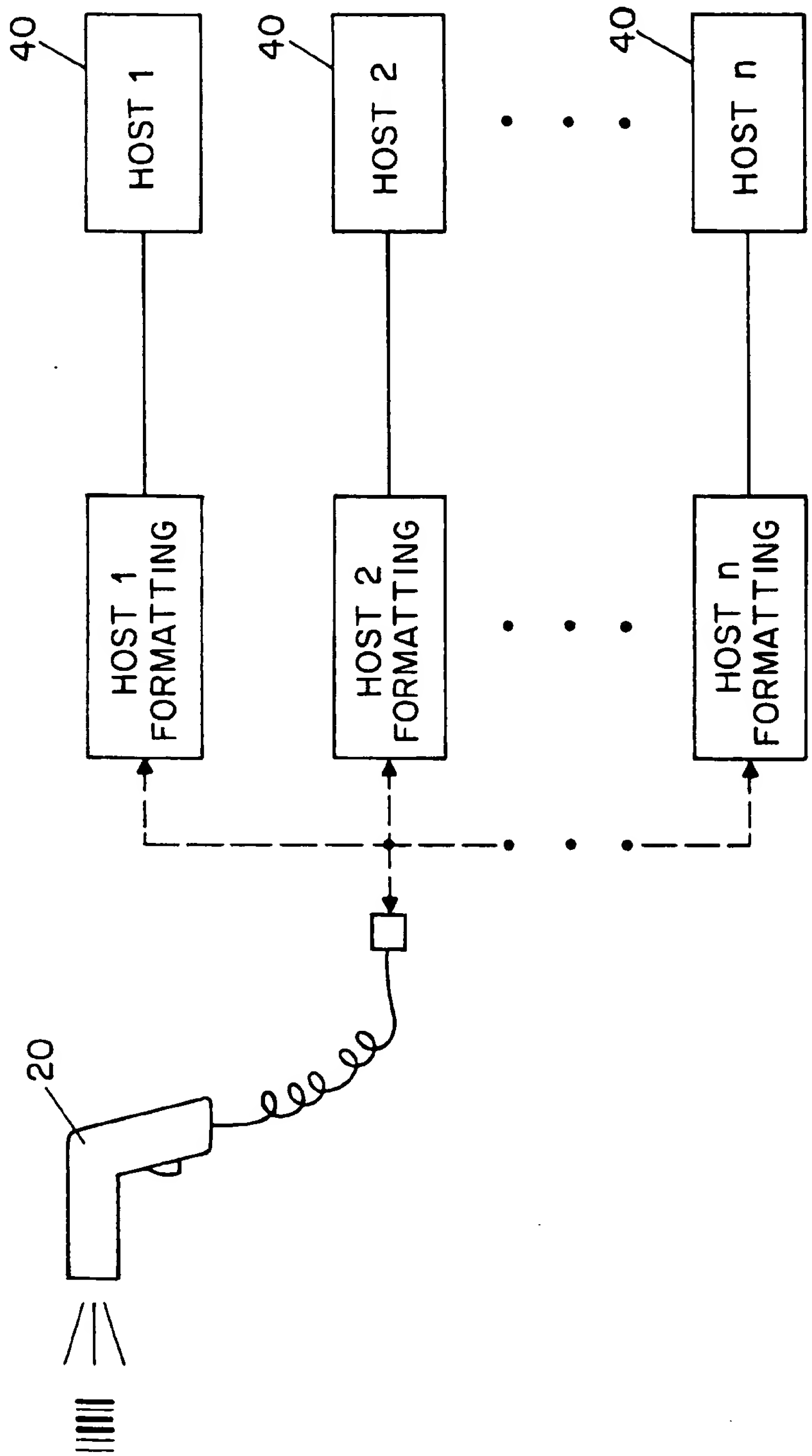

40
HOST 1
40
HOST 2
40
HOST n
HOST 1 FORMATTING
HOST 2 FORMATTING
HOST n FORMATTING
20

FIG. 4

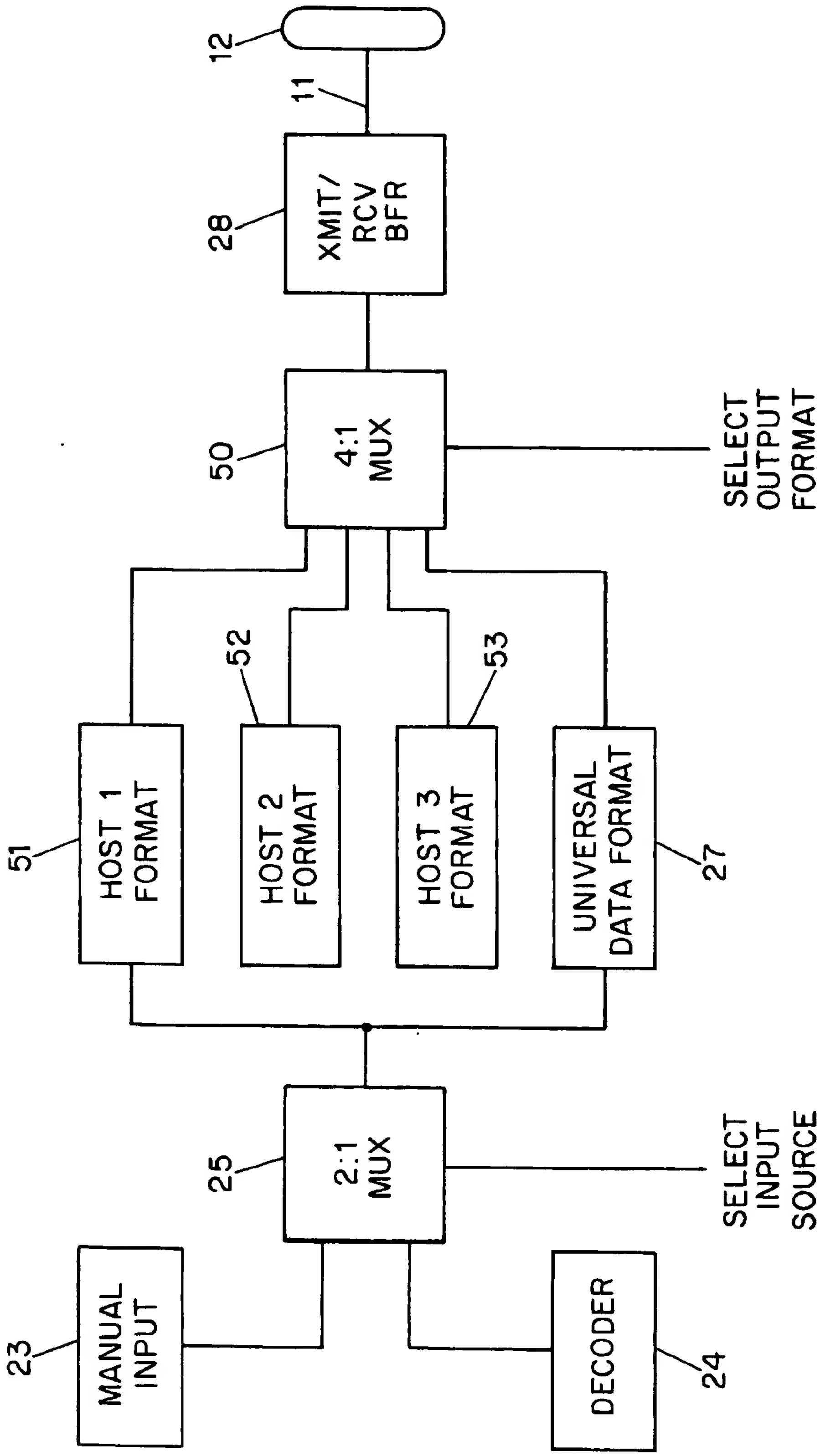


FIG. 5

10/29/2003, EAST Version: 1.4.1

UNIVERSAL HOST INTERFACE FOR DATA ACQUISITION SYSTEMS

This application is a continuation of application Ser. No. 08/439,833, filed May 12, 1995, abandoned.

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates to a universal host interface system for data acquisition devices such as bar code scanners and in particular to such a system which implements a programmable cable interface device to couple a universal and generic data acquisition device interface to a specific host device.

2. Description of Related Art

Various types of data acquisition devices exist which allow a user to collect and transmit data to a host computing device. Data acquisition devices may be terminals, with keypad entry means and a display. In addition, data acquisition devices may be bar code scanners in a stand alone configuration or integrated with a terminal. Various types of bar code scanning devices are known in the art which optically scan a bar code printed on a substrate for capture of data encoded in the bar code. Laser bar code scanning devices use a laser light source such as a visible laser diode to scan the bar code by sweeping the laser source across the bar code with a mechanically oscillating reflecting element such as a mirror. The light reflected from the bar code is collected by a photosensor directly from the bar code or retroreflectively off of a portion of the scanning mirror. The analog waveform output by the photosensor has an amplitude which represents the amount of laser light reflected off the bar code. Thus, the varying amplitude of the analog waveform represents the relative spacing of the various bars and spaces of the bar code, which in turn represents the data encoded therein. The analog signal is processed, digitized and decoded into data representative of that which had been encoded into the target bar code.

Bar code scanning devices are also known in the art which are based on solid state imagers such as charge coupled devices (CCDs). CCD based bar code readers are either one-dimensional or two-dimensional. One-dimensional CCD bar code scanners use a linear array of photosensors to capture an image of a cross section of the entire linear bar code at once and produce an analog waveform whose amplitude is representative of the darkness and lightness of the bars and spaces of the captured image. The electric charge stored in each element of the CCD array as a function of the amount of light sensed by an area covered by each element is shifted out serially to form electric signals for further processing, digitizing and decoding. Two dimensional CCD bar code readers operate similarly to capture an image of an entire two-dimensional bar code symbol at once and process it accordingly. The waveform is digitized and decoded in a means similar to laser scanners.

The decoded signals are typically transmitted to a host device for processing, storage, and the like. Many types of host devices exist, depending on the particular application desired by the user. For example, scanners are used at POS cash register terminals to scan in the bar code of a product, where the terminal uses the bar code data as a pointer to look up the price and item description in memory. Scanners are also used to interface to personal computers with keyboard wedges, where the scanner is inserted in the path between the keyboard and the computer, and the scanner input must be configured to appear the same to the computer as keyed

input from the keyboard. Scanners are also used to drive RS-232 computer interfaces in other applications.

In most cases, each different type of host device implements a different type of input/output interface, thus requiring different mechanical connectors, different electrical and data formats and protocols, etc. depending on the user's application. Prior art devices were thus usually configured by the manufacturer to interface with one or more host devices. This custom type of scanner-host device matching is costly and inefficient.

In U.S. Pat. No. 5,258,604, a system is suggested which implements a bar code scanning device which is configured to accept any of a plurality of differently configured interface boards in its handle, wherein each type of interface board is specially adapted to mate with an associated host device. The decoder in the scanner polls the interface board upon power-up and reads an identification code from the particular interface board which the user has inserted. The identification code is used by the decoder to access configuration and formatting data from an on-board memory in order to transmit data to the interface board and ultimately to the host device in the required manner. This requires a user to have to swap electrical boards inside the scanner device whenever he desires to use the scanner with a different host device, which is unwieldy and potentially damaging to the boards due to electrostatic discharge (ESD) phenomena. In addition, when a new type of host device is desired to be used with a scanner already in use in the field, the scanner must be physically reconfigured in order to be able to recognize a new identification code from a new type of interface board and to properly format the decoded data for transfer to the new type of host device. Thus, this type of data acquisition device system is not readily upwards compatible with new host devices with which it may be desired to be connected.

It is also desired in the field of data acquisition such as bar code scanning to be able to use a particular data acquisition device with different host devices at different times by simply connecting the data acquisition device to the host device in a "plug and play" environment, without having to physically reconfigure the data acquisition device with different internal interface boards as in the prior art. This may occur, for example, with a data acquisition device that is used to scan bar codes during the daytime at a POS terminal, and then is used at night time for inventory collection purposes in a storage room with a different host device. Further, even if the two host devices are of the same type and configuration (thus using the same interface), the user may operate the scanner and/or host devices in different ways depending upon the location. Rather than having to reprogram the scanner every time a different host is connected (usually by scanning in various "parameter bar codes"), it is desirable for the scanner to retain its operating parameters and implement them automatically in accordance with the connected host device or in accordance with the scanning of a "location defining" bar code.

It is therefore an object of the present invention to provide a modular and universal data acquisition system which overcomes these and other problems of the prior art.

It is a further object of the present invention to provide a modular and universal data acquisition system in which the data acquisition device which does not need to be reconfigured in order to communicate with different host devices.

It is a further object of the present invention to provide such a system in which the data acquisition devices are upwards compatible and will communicate with newly designed host devices without the need for reconfiguration.

It is a further object of the present invention to provide such a system in which the host devices are upwards compatible and will communicate with newly designed data acquisition devices without the need for reconfiguration.

It is an even further object of the present invention to provide such a system in which the data acquisition device may easily and conveniently be used with any number of differently or similarly configured host devices without the need for the user to manually change operating parameters depending on the host device to which it is connected.

It is further object of the present invention to provide such a system which advantageously implements an intelligent cable interface device in order to provide communications formatting and protocol between an associated host device and any type of data acquisition device.

SUMMARY OF THE INVENTION

In accordance with these and other objects, provided is a universal interface system for use with a data acquisition system including a data acquisition device for providing digital data signals indicative of acquired data for transmission to a host device, and a host device having host-specific input/output data format requirements. The universal interface system comprises universal data exchange means located in the data acquisition device for providing digital data signals in a universal data exchange format independent of the data format requirements of the host device; and a host interface module coupled to the data acquisition device and to said host device, ~~for transmitting acquired digital data from the data acquisition device to the host device, the interface module comprising means for translating digital data from the universal data exchange format to the host-specific input/output data format requirements.~~ The interface module comprises a buffer for storing a host parameter data word used in communications with the host device. The universal data exchange means comprises a memory for storing a plurality of host parameter data words, each of the host parameter data words being associated with one of a plurality of different host interface modules; and means for reading from the memory and transmitting to the host interface module the host parameter data word associated with the particular host interface module operatively connected thereto.

The data acquisition device comprises means for entering host parameter data words by a user. In embodiments wherein the data acquisition device comprises a bar code scanner capable of electro-optically reading bar code symbols encoded with bar code data and providing a digital bar pattern representative of bars and spaces comprising said bar code symbol, the bar code scanner comprising a decoder for converting said digital bar pattern into said digital data signals, the decoder of the invention further comprises means for determining if the digital bar pattern is representative of a host parameter data word, and means for storing in the memory the host parameter data word for subsequent transfer to the host device.

The system of the present invention undergoes an initialization routine wherein the universal data exchange means requests an identification code from the host interface module, the host interface module provides in response thereto an identification code stored therein, the universal data exchange means utilizes the provided identification code to read from said memory the host parameter data word associated with the identification code, the universal data exchange means transmits to the host interface module the host parameter data word read from said memory, and the host interface module overwrites the default mode with the host parameter data word provided by the universal data exchange means and operates in accordance with the host parameter data word.

BRIEF DESCRIPTION OF THE DRAWINGS

FIGS. **1**(*a*) and **1**(*b*) are a block diagram of the preferred embodiment system of the present invention;

FIG. **2** is a pictorial diagram of the system of FIG. **1**;

FIGS. **3**(*a*) to **3**(*e*) are functional diagrams illustrating the data exchange protocol of the system of FIG. **1**;

FIG. **4** illustrates an application of the system in which a data acquisition device may connect with a plurality of host interfaces; and

FIG. **5** is a block diagram of an alternative embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

The preferred embodiment of the present invention will now be described in detail. FIGS. **1**(*a*), **1**(*b*) and **2** illustrate the preferred embodiment data acquisition system **10** which comprises a data acquisition device **20**, a host interface cable **30**, and a host device **40**. The host device **40** may be any type of computing device known in the art which is configured to accept data input in any type of format, such as a computer with an RS-232 communications port, an Optically Coupled Interface Adapter device (OCIA), a keyboard wedge configuration (as in a personal computer), or an IBM 4683 POS device, any of which will accept bar code data obtained when the data acquisition device **20** implements bar code scanning (or acquires data by other means such as magnetic stripe reading, keyed input, etc.).

The data acquisition device **20** is configured in the present invention in a generic manner, and has a standard interface protocol which is referred to herein as SDCI (Standard Data Communications Interface). The data acquisition device **20** in the preferred embodiment is a scanner integrated terminal and comprises a bar code scan engine **22** and a decoder **24**. The bar code scan engine **22** may be of any type known in the art as described above which produces a digitized signal (digital bar pattern DBP) which is representative of the relative spacing of the bars and spaces of the bar code which is scanned. The decoder **24** may also be of any type known in the art which operates on the digital bar pattern via various algorithms stored in an associated memory in order to decode the digital bar pattern to a digital data signal in accordance with the particular symbology being implemented. The decoder **24** may be configured to decode bar code symbols in any one symbology desired, or it may be configured in an autodiscrimination mode to automatically detect and decode bar codes of more than one symbology and provide the digital data accordingly. A manual input device **23** such as a keypad is provided, along with the decoder **24** output, to a multiplexer **25** which provides the selected data signal to a universal data formatting means **27**. The universal data formatting means **27** uses the digital data from the selected source and formats it in the SDCI protocol, to be described below, for output via a transmit/receive buffer **28**, which performs electrical communication with the host device **40** via the host interface cable **30**, which is coupled to a data acquisition device cable **11** and data acquisition device connector **12**. Importantly, the data transmitted by the buffer **28** is in the universal, non-host specific SDCI protocol which is common to all types of data acqui-

sition devices recognized by the data acquisition system **10**. Thus, any type of data acquisition device **20** configured in the SDCI protocol will be usable in the system **10** without having to perform any type of reconfiguration, reprogramming or the like. This "plug-and-play" mode allows users to easily switch data acquisition devices within a given system, in accordance with their needs, and it allows upwards compatibility with new data acquisition devices as they are designed and manufactured in accordance with the universal SDCI protocol.

A parameter memory **26** comprises non-volatile electrically reprogrammable memory space (e.g. a non-volatile RAM) for storage of certain operating parameters of both the data acquisition device **20** as well as the host interface cable **30**, as will be further described below. These parameters may optionally be programmed by the user by entering either manually or by scanning a dedicated parameter bar code from a menu provided with the system **10**. By scanning an operating parameter bar code, preset default values for the specified functions are overridden. For example, if the user wishes to instruct the data acquisition device to use a soft beeper tone rather than the preset default tone, he may scan in a preprinted parameter bar code provided with the system, which is detected by the decoder to be an operating parameter rather than data and which is then routed to and stored in an appropriate location in the parameter memory **26**. This parameter will control operation of the tone generator used to signal the user when a bar code has been detected. If the user wishes to again change this parameter (e.g. increase the volume of the tone), he simply scans in an appropriate preprinted bar code and the parameter is updated in the memory **26** for subsequent use.

In addition to storing various data acquisition device parameters, the parameter memory **26** may be used to store optional parameters relating to the operation of the host interface cable **30** and the host device **40**. Importantly, these parameters are only optional, and are not required to be used in order for the data acquisition device **20** and the host device **40** to communicate with each other. That is, a set of default values, to be described below, is preset in the interface cable **30**, which controls the operation of the system communications without requiring operator programming.

The parameter memory **26** is capable of storing parameter data in individual locations for each type of host device **40** which may be connected thereto. A host parameter search and read means **21** is used by an initialization process to search through the memory **26** to look for parameter data matched to its associated host **40**. In this way, an interface cable **30** may access specific user-defined parameters in order to override system defaults set in the cable **30**. As a result, the data acquisition device **20** may be used with any type of host/cable system, and the particular interface cable **30** to which it is connected will fetch from the parameter memory **26** the operating parameters with which it is associated. The data acquisition device **20** thus functions as a "dumb" memory device for storage of the host parameters. That is, the data acquisition device **20** need know nothing about the effect of the host parameters which it downloads to the interface cable **30**.

The functions of the universal data formatting means **27** and the host parameter search means **21** may be advantageously implemented by an embedded system such as a microprocessor device programmed to perform such functions. Such implementation is known in the art and need not be described in detail.

A power down circuit **29** controls the operating power used by the components in the data acquisition device **20** and will lower the operating power requirements when instructed by a timeout sense circuit **29***a* or by command from the host device **40** as will be described below. This allows the data acquisition device **20** to enter a "sleep" mode whenever no data has been acquired in a certain timeout period (e.g. 3 minutes) or when desired by the user of the host device **40**. The sleep mode is exited whenever data is input by the acquisition device **20** or upon command by the host device **40**.

The interface cable **30** functions as the host-specific link from the universal SDCI output of the data acquisition device **20** to the particular host device **40** with which it is associated. Each different type of host device **40** will require the use of a differently configured interface cable **30** in order to be able to communicate properly with a data acquisition device **20**. The interface cable **30** comprises an SDCI bus **32** coupled to a processing board **36**. The processing board **36** is embedded within a plastic housing **31** as shown in FIG. 2. The interface cable **30** also comprises a host link **34**, which is coupled to the processing board **36** and has an interface/host connector **33** which is designed to mechanically and electrically mate with an associated host connector **42** found on the host device **40**.

In operation, the data acquisition device **20** is coupled to the interface cable **30** by inserting the data acquisition device connector **12** into the mating interface/device connector **35**, and the interface cable **30** is coupled to the host device **40** by inserting the interface/host connector **33** into the mating host connector **42**. The interface cable **30** is configured, by appropriate circuitry within host format and protocol translation means **39**, to communicate with the particular host device **40** to which it is connected. Functionally, the interface cable **30** comprises the host format and protocol translation means **39**, which accepts as input via transmit/receive buffer **37** universal SDCI data from the data acquisition device **20**. A host parameter buffer **38** stores host parameter data fetched during power up initialization from the data acquisition device **20**. The host format and protocol translation means uses the data from the parameter buffer **38** to appropriately configure data in the format required by the host device **40**. In particular, upon power-up, the interface cable **30** transmits upon request a host ID code from a host ID buffer **43** to the data acquisition device **20** via the SDCI protocol. The data acquisition device **20** receives the host ID code and searches the parameter memory **26** to see if the user had previously entered application-specific parameters. When the system is powered up for the first time, the parameter memory **26** will be cleared (all data bits set to zero), and the system **10** will operate in a preprogrammed default mode. As such, the system needs no information from the data acquisition device **20** in order to perform communications between the data acquisition device **20** and the host device **40** since the host-specific interface cable **30** will be matched to the particular host device **40** in use via the host format and protocol translation means.

When the user desires to change any or all of the preprogrammed default values for operating parameters of the interface cable **30**, he does so for example by scanning a preprinted parameter bar code accompanying the system **10**. As with scanner parameters previously described, the decoder **24** detects that the bar code scanned is a parameter bar code and not data, and the decoder **24** routes the scanned parameter along with a host identification tag (e.g. HOST 001) to an empty location in memory **26**. The entire set of host parameters is also transmitted to the host interface cable **30** and stored in the buffer **38** for subsequent use. In the

preferred embodiment, the buffer **38** is a volatile memory (i.e. RAM) in order to reduce cost.

The next time the system **10** is powered up, the interface cable **30** upon request sends its host ID code to the data acquisition device **20**, which is used in a search of the memory **26** for any host parameters which had been previously entered and stored. When a match is found between the host ID code and a location in memory **26**, the parameter data is fetched and sent to the interface cable **30** via the SDCI bus **32** for storage in the parameter buffer **38**. Thus, once the user has reconfigured the operation of the interface cable **30** with a particular data acquisition device **20**, the system **10** operates in the same fashion every time until the user re-programs the parameters.

Advantageously, as shown in FIG. **4**, the user may utilize the data acquisition device **20** with a different host device **40** at, for example, a different location in the workplace. The host parameter memory **26** will store the user-scanned parameters in a different location, and the interface cable **30** of the second system will send its own unique ID code to request its parameters. Thus, the data acquisition device **20** can be taken from system to system, wherein each host device has an associated interface cable **30** connected thereto, and the data acquisition device will instruct the interface cable **30** to operate in the manner desired by the user with the user only having to scan a parameter bar code only once for each particular system.

The system **10** of the present invention advantageously implements a set of default parameters wherein each default value is set to logical 0. The memory **26** stores multiple parameters in the same location for a given interface cable **30**. By scanning a particular parameter bar code, only that parameter is changed and the remaining parameters are left at the default of 0. By using 0 as defaults, the data acquisition device **20** does not have to store default tables as was required in the prior art. Additionally, the device **20** will "know" the defaults for all cables designed and used in the future. That is, by using a universal default parameter setting of all 0's, data acquisition devices may be designed to operate in conjunction with currently existing cables and, similarly, cables in the future may be designed to operate with existing data acquisition devices.

The SDCI protocol implemented between the data acquisition device **20** and the host interface cable **30** in the preferred embodiment of the present invention will now be described in detail. Communications between the data acquisition device **20** and the host interface cable **30** are carried out on a two-line serial interface. Thus, no matter which host-specific cable **30** is used, the communications between the data acquisition device **20** and the host interface cable **30** are the same. This is made possible by the flexibility of the parameter structure and the host directives to be described herein. All data sent between the data acquisition device **20** and the host interface cable **30** are done with one of the following opcodes.

The following Table A is a list of all the opcodes supported by the SDCI protocol in the preferred embodiment of the present invention. Table A identifies the source device (data acquisition device **20** or interface cable **30**) allowed to send a message of that type. Opcodes designated with the type "cable" (C) means that only the cable **30** can transmit that set of opcodes. Similarly, "data acquisition" (D) type opcodes can only be transmitted by the data acquisition device **20**, and "cable/data acquisition" (C/D) types can be transmitted by both the cable **30** and the data acquisition device **20** as required.

TABLE A

SDCI OPCODES

Name	Class	Source	Description
ADF_DATA	Data	D	Acquired data in ADF packet format
ATTENTION	Control/Status	C	Inform data acquisition device of cable presence
BEEP	Control/Status	C	Request the data acquisition device to sound the beeper
BEEP_DIRECT	Control/Status	C	Tell Data Acquisition device how beeper parameters should be set
CMD_ACK	Control/Status	C/D	Positive acknowledgment of received packet
CMD_NAK	Control/Status	C/D	Negative acknowledgment of received packet
DCU_DATA	Data	C/D	Data Acquisition Device Customization Utility Parameter Buffer
DCU_REQUEST	Control/Status	C/D	Request Data Acquisition Device to Send all its parameters out as DCU_DATA
DECODE_DIRECT	Control/Status	C	Tell Data Acquisition Device how decode parameters should be set
ID_REPLY	Control/Status	C	Send cable ID string to Data Acquisition Device
ID_REQUEST	Control/Status	D	Ask Cable/Host for its ID string
LEDOFF	Control/Status	C	Request the data acquisition device to deactivate LED output
LEDON	Control/Status	C	Request the data acquisition device to activate LED output
PARAM_DATA	Data	D	Anonymous Parameter Data
POLL	Control/Status	D	Confirms Cable resence after configuration
RESET	Control/Status	D	Requesting the cable to start the configuration process
SCALE_DATA	Data	D	Data from weight scale via data acquistion device
SCALE_REQUEST	Control/Status	C	Request to Data Acquisition Device to send weight scale data
SCAN_DISABLE	Control/Status	C	Prevent the operator from scanning bar codes
SCAN_ENABLE	Control/Status	C	Permit the operator to scan a bar code
SLEEP	Control/Status	C/D	Request to place the data acquisition device or cable into low power
START_DECODE	Control/Status	C	Tell Data Acquisition Device to attempt to

TABLE A-continued

SDCI OPCODES			
Name	Class	Source	Description
			decode a bar code
STOP_DECODE	Control/Status	C	Tell Data Acquisition Device to abort a decode attempt
TYPE_LEN_DIRECT	Control/Status	C	Tell Data Acquisition Device how to set its Code Type and Length parameters

Note:
D = Data Acquisition Device,
C = Cable,
C/D = Cable/Data Acquisition Device

The cable **30** and the data acquisition device **20** exchange the messages in Table A in data packets. A packet is a collection of bytes framed by the proper SDCI protocol formatting bytes. The maximum number of bytes per packet allowed by the SDCI protocol for the transmission of data to the cable for any transaction is specified by the cable in the ID_REPLY message as described below. The following Table B is the general packet format for SDCI messages, and descriptions of fields that occur in all messages.

TABLE B

Packet Format
<PAL><OPCODE><MESSAGE SOURCE><STATUS><DATA AND OTHER PARAMETERS><CHECKSUM>
Field Descriptions

Field Name	Format	Sub-Field	Meaning
PAL (Protocol and Length Field)	Bits 0 to 6	Length	Length of message not including the check sum bytes.
	Bit 7	Class	0 = data 1 = control/status
OPCODE	1 Byte	See Table A	Identifies the type of packet data being sent
MESSAGE SOURCE	1 Byte	0 = Data Acquisition Device 3 = Cable	Identifies where the message is coming from
STATUS	Bit 0	Retransmit	0 = First time packet is sent 1 = Subsequent transmission attempts
	Bit 1	Continue	0 = Last packet 1 = Data continued with next packet(s)
	Bit 2	Low battery power	
DATA AND OTHER PARAMETERS	Variable number of bytes		The data sent in the field varies in accordance with the particular message
CHECKSUM	2 Bytes	2's complement sum of message contents excluding checksum	Checksum of message

Major SDCI Messages and Transactions

The data transactions which occur among the data acquisition device **20**, the interface cable device **30**, and the host device **40** are now described in detail. As used herein, the term "packet transmitter" refers to the device which is currently transmitting a data packet, and the term "packet receiver" refers to the device which is currently receiving a data packet. For a given data transaction, the appropriate opcodes set forth in Table A are exchanged among the aforementioned devices as shown in FIGS. **3**(*a*) through **3**(*e*). In addition, for each opcode sent, the CMD_ACK and CMD_NAK opcodes (positive acknowledgment of received packet and negative acknowledgment of received packet, respectively) are sent by the packet receiver to the packet transmitter to indicate the status of the received data packet. The CMD_ACK opcode is sent by the packet receiver when the received data packet passes a checksum test and the receiver successfully processed the data. Likewise, the CMD_NAK opcode is sent by the packet receiver when the received data packet fails the checksum test or some other error occurred while handling the message.

Power-up Initialization

FIG. **3**(*a*) illustrates the initialization data transaction implemented by the SDCI system. Upon power-up, the ID_REQUEST opcode is sent by the data acquisition device **20** to the interface cable **30**, which directs the interface cable **30** to send a unique identification string which is stored in its host ID buffer **43**. This unique cable/host ID is used to identify the particular cable/host to which the data acquisition device is communicating. The <DATA AND OTHER PARAMETERS> field is empty in the ID_REQUEST opcode since this opcode is simply a command to the cable **30**.

Upon successful receipt of ID_REQUEST, the cable **30** will respond by sending the unique ID in an ID_REPLY opcode. The unique ID is fetched from the host ID buffer **43** and inserted into the <DATA AND OTHER PARAMETERS> field. Also inserted in the <DATA AND OTHER PARAMETERS> field are other system parameters such as the cable release version (to ensure software compliance),

the maximum massage length accepted by the cable **30** (including the checksum), the minimum wait time that the data acquisition device **20** should wait for the cable to issue a CMD_ACK or CMD_NAK response to an ADF_DATA message, the output decode type (for data acquisition devices with bar code scanning capabilities) which specifies if the data should be in intermediate scan data (binary image of bar code) or ASCII format, a flag to indicate whether the cable **30** will support continued ADF_DATA messages for long data transmissions, and the size of the cable's anonymous parameter buffer.

Upon receipt of the ID_REPLY opcode from the cable **30**, the data acquisition device **20** will use the unique ID word and search the parameter memory **26** to see if parameters have been previously stored for that particular cable/host configuration. If the data acquisition device finds a match and thus has parameters to send to the cable **30**, it will load them into the <DATA AND OTHER PARAMETERS> field of a PARAM_DATA message and send it to the cable **30**. If no match is found, indicating that no parameters have been loaded for that particular cable/host configuration, then a set of default values (i.e. all zeroes) will be sent in the PARAM_DATA message.

Although not illustrated in FIG. **3**(*a*), the RESET and ATTENTION opcodes, which are described in more detail further below, are instrumental in ensuring a proper initialization handshake sequence occurs.

Transfer of Acquired Data

The ADF_DATA (Advanced Data Formatting) opcode is used to transmit data entered into the data acquisition device **20**. When the data acquisition device **20** comprises bar code scanning capabilities as in the preferred embodiment system shown in FIG. **1**, this opcode is used in order to transport decoded bar code data to a host device **40** via the interface cable **30**. Referring to FIG. **3**(*b*), data is captured by the bar code scan engine **22**, decoded by the decoder **24**, and sent via the ADF_DATA message to the cable device **30**. Upon receipt of this message, the host format and protocol translation means **39** in the cable **30** will format the data in conjunction with the host parameter buffer **38** for transmission to the host device **40** and send the data to the host. If the transfer of data to the host is successful, a CMD_ACK is sent to the data acquisition device **20**; otherwise a CMD_NAK is sent that indicates what failed.

The <DATA AND OTHER PARAMETERS> field of the ADF_DATA message contains data which identifies the bar code type being sent as well as the decoded data itself. The format of this field is:

<data type><length><data><data type><length> <data> . . . <EOB>

The ADF data field must end with a zero <EOB> byte which indicates the end of the ADF data stream.

In particular, in the ADF protocol, the bar code type indicates the particular symbology of the decoded data, such as EAN 8, EAN 13, Code 39, Codabar, UPC, Code 128, etc. The data type field indicates whether the transmitted data is ASCII, ALT-Shifted, Numeric, Extended Keypad, etc., as required by a particular system application. When the data acquisition device **20** needs to send an ADF_DATA message that is longer than the length that the cable will accept, then the data is sent in smaller packets. If an ADF_DATA message is going to be continued by one or more messages, then the "continue" bit should be set in the status field (see Table A above). The continue bit shall be set for all ADF data packets that make up the message until the last piece, which has the continue bit cleared. This indicates that the continued message is complete with the last packet. When a cable **30** has indicated in its ID_REPLY message that it does not accept continuations of data packets, then the data acquisition device **20** does not send any ADF_DATA to the cable if it cannot fit the entire data packet in one ADF_DATA message.

Cloning of Data Acquisition Device Parameters

The system of the present invention may be advantageously used in order to effect "cloning" of data acquisition device parameter configurations without the need for an operator to manually program, by keypad entry, bar code menu scanning or the like, data acquisition device parameters in each individual device **20**. Rather, an operator may program parameters in one data acquisition device **20** and allow the parameter memory to be uploaded to a host device, where it is temporarily stored for subsequent download to a different data acquisition device **20**. This a time-saving implementation of the present system which ensures that similar devices will be programmed identically without a chance for operator error from device to device.

FIG. **3**(*c*) illustrates the protocol sequence implemented by the system in order to effect cloning of the data acquisition parameters. In this case, the parameters are scanner parameters which relate to the operation of the decoder **24** of a data acquisition device **20** having bar code scanning capabilities as shown in FIG. **1**(*a*). Referring to FIG. **3**(*c*), when the host device **40** requires the transfer of scanner parameters from the data acquisition device **20**, the host will send a DCU_REQUEST (Data Acquisition Device Customization Utility Request) message to the interface cable **30**. The cable **30** translates this message to SDCI format and sends DCU_REQUEST to the data acquisition device **20**. The <DATA AND OTHER PARAMETERS> field in the DCU_REQUEST message specifies a request for either an image of the scanner parameters or an image of the default parameters, as required.

Upon receipt of the DCU_REQUEST, the data acquisition device **20** will load the requested parameters from the memory **26** into the <DATA AND OTHER PARAMETERS> field of a DCU_DATA message for transmitting to the cable device **30**. In particular, the <DATA AND OTHER PARAMETERS> field of DCU_DATA specifies the revision level of the parameter buffer in order to ensure compatibility when cloning, the location within the parameter buffer where the image begins, and the data from the parameter buffer. DCU_DATA is received by the cable **30**, translated to the format suitable for communication with the particular host **40**, and sent to the host **40** for storage in a buffer **45**.

When a second data acquisition device **20** is connected to the host/cable assembly for cloning of uploaded parameters, the host device **40** initiates the transfer of scanner parameters to the data acquisition device **20** by sending a DCU_DATA message to the interface cable **30**. The DCU_DATA message has in its <DATA AND OTHER PARAMETERS> field the scanner parameters obtained from the initial upload. The cable **30** translates this message to SDCI format and sends DCU_DATA to the data acquisition device **20** for storage in the scanner parameter memory **26**. This process can be repeated with other data acquisition devices **20** in the same fashion.

If the revision level of the parameter buffer does not match the revision level of the data acquisition device **20**, then the parameter data is not written into the nonvolatile memory therein (the transfer is not completed).

In the alternative, the cloning operation described herein may be effected by the data acquisition device **20** and the

cable 30 without needing the host device 40. In this instance, the user scans a predetermined "upload" bar code which is interpreted by the decoder as a command to transmit DCU_DATA to the cable device 30. The parameter data is stored in a temporary buffer in the cable 30 rather than in the host device 40 as mentioned above. The user may subsequently connect the interface cable 30 to a new device 20 and scan a second "download" bar code which is interpreted by the decoder as a command to request DCU_DATA from the temporary storgae buffer of the cable 30. The stored parameter data is sent to the data acquisition device 20 in the manner mentioned above to effect completion of the cloning operation without the need to be connected to a host device 40.

Scale Data Exchange

In the preferred embodiment system, a data acquisition device may have connected thereto a scale 60 for weighing items and providing scale data indicative of the weight to the host device 40. Referring to FIG. 3(*d*), the host device 40 will request scale data by issuing a request to the interface cable 30, which translates the request and sends a SCALE_REQUEST command to the data acquisition device 20. The data acquisition device 20 will read the attached scale device 60 and respond with a SCALE_DATA message. The cable 30 will translate the scale data from the SDCI format to the format required by the host device 40.

The <DATA AND OTHER PARAMETERS> field in the SCALE_REQUEST command issued by the cable device 30 indicates to the data acquisition device 20 and scale 60 the amount of time to wait for the scale to settle prior to obtaining a measurement, the units of measure (i.e. pounds or kilograms), and data which may be displayed on a display device of the scale 60. The <DATA AND OTHER PARAMETERS> field in the subsequent SCALE_DATA reply message indicates scale status (hardware error, scale not settled in required time, units of measure, weight of item over limits of scale, and/or weight of time too small to measure) and the measurement data in ASCII format.

Other SDCI Messages and Transactions

In addition to the major SDCI transactions and associated opcode messages described above, the following SDCI message opcodes are used in the preferred embodiment system.

ATTENTION

The ATTENTION opcode has a null <DATA AND OTHER PARAMETERS> field and is sent by the cable 30 during the initialization process to announce its presence to the data acquisition device. If the ATTENTION/ID_REQUEST/ID_{13} REPLY/PARAM_DATA/ process fails due to communications problems, the cable 30 will send ATTENTION repeatedly until ID_REQUEST is received. No matter what state the data acquisition device is in, the receipt of an ATTENTION message from the cable 30 will cause the data acquisition device to follow this sequence. After the data acquisition device receives the ATTENTION message, the data acquisition device will ask the cable to identify itself by an ID_REQUEST.

From the time that the data acquisition device has reset or it has received an ATTENTION message, the data acquisition device must require that the entire initialization process occurs in sequence without any errors. If any error occurs in the receipt of ATTENTION, ID_REPLY, in the transmission of RESET, ID_REQUEST or PARAM_DATA, or in the transmission of an ACK to the cable, then the data acquisition device must send a RESET to the cable. If this transmission fails, then the data acquisition device will continually send RESETs until acknowledged with an ATTENTION.

If the initialization process does not complete within two seconds, then the data acquisition device must sound a beep that identifies a problem with the cable 30. This beep should be repeated every two seconds until the problem is fixed.

BEEP

The BEEP opcode has a beep code field in the <DATA AND OTHER PARAMETERS> section. This opcode is intended to be used with those bar code scanning data acquisition devices having decoders equipped with a beeper; it instructs the decoder to sound the beep sequence indicated by the beep code byte field. The decoder should beep the sequence provided in the BEEP opcode, only if the beep sequence is supported by the decoder, and the sounding of the beep specified does not interfere with the decoder's user interface. If the decoder does not support a beeper, then this message can be ignored. The beep code field indicates, for example, a high pitch, low pitch, or combination of both as required by a particular system.

BEEP DIRECT

The BEEP_DIRECT opcode sets the frequency, duration and volume parameters in its <DATA AND OTHER PARAMETERS> section. A status field can also indicate a permanent storage bit indicating a permanent change. The decoder supports the BEEP opcode for all frequency values that it can support. The decoder shall store directive information to non-volatile storage when the permanent storage bit is set. The duration parameter is supported by decoders that have programmable durations for decode beep. Beep volume is supported by decoders that support volume setting as closely as possible.

DECODE DIRECT

This opcode is used by the cable 30 to instruct the data acquisition device 20 how decode parameters should be set, as specified by the host device 40. When a cable 30 receives directives from the host device 20 pertaining to these parameters, the cable 30 sends the directives to the decoder by using this message. The directives are transmitted in the <DATA AND OTHER PARAMETERS> field and include bar code decoding redundancy parameters, UPC security level, linear security level, laser-on timeout, laser-off timeout, laser sleep time, motor sleep time, LED blink frequency, double touch indication, and ASCII stitching.

LED OFF, LED ON

The cable 30 sends these opcodes to the decoder to turn off and on the decode LED, respectively. These opcodes have null <DATA AND OTHER PARAMETERS> fields.

POLL

This command is useful when the data acquisition device 20 has a separate power source from the cable 30, and the decoder wishes to check regularly that the cable 30 is still connected. The cable 30 must respond to the POLL message within 2 seconds, or the decoder can assume that the cable is not plugged in. The decoder may send a POLL to the cable 30 at any time, except when awaiting an acknowledgment from a previously sent message. If the cable does not respond within 2 seconds, then the decoder shall send a RESET. This opcode has a null <DATA AND OTHER PARAMETERS> field.

RESET

This opcode is sent by the decoder when a communications line failure is detected (i.e. no response over a set amount of time, or all resend attempts fail). Upon receipt of the RESET message, the cable 30 stops what it is doing and sends an ATTENTION. Upon sending RESET, the decoder continually sends RESETs at least every half second, until the cable responds with ATTENTION. This opcode has a null <DATA AND OTHER PARAMETERS> field

SCAN DISABLE, SCAN ENABLE

The cable **30** can disable and enable bar code scanning, respectively, to a data acquisition device **20** equipped with bar code scanning capability. When the decoder receives the SCAN_DISABLE opcode, it will not transmit to the host device until a subsequent SCAN_ENABLE opcode is received. At startup, the decoder assumes SCAN_ENABLE. These opcodes have null <DATA AND OTHER PARAMETERS> fields.

SLEEP

The SLEEP opcode can be sent by either the cable **30** (under command of the host device **40**) or the data acquisition device **20** to tell the receiver to go into a low power mode. The host device **40** can instruct the cable device **30** to go to sleep, in which case the cable **30** goes into a low power mode and sends a SLEEP command to the data device. In addition, the data device **20** can sense via the timeout sense means **29***a* that data has not been collected for a certain period of time. This will put the data acquisition device **20** into a low power mode, and the data acquisition device **20** will then instruct the cable to do the same by using a SLEEP command. In either case, the system is waked up by the data acquisition device **20** sensing an attempt to input data (e.g. a trigger pull if the device **20** comprises a bar code scanner). When the cable is in sleep mode, and the decoder tries to send the cable any message, then the cable will wake up and resume operation where it left off or respond with an ATTENTION message. This may involve no more than two resend attempts on the decoder's part. Decoders that support a low power operation are required to sleep when this command is received. This opcode has a null <DATA AND OTHER PARAMETERS> field. There is no initialization process after a wake-up as there is after a cold start of the system.

START DECODE, STOP DECODE

This opcodes instruct a data acquisition device **20** equipped with bar code scanning capability to start and stop (if not yet timed out) a scan and decode session, respectively. These opcodes is interpreted by the device **20** as a trigger pull and release, respectively, or other scan initiation (i.e. object sensing). These opcodes have null <DATA AND OTHER PARAMETERS> fields.

TYPE LEN DIRECT

The <DATA AND OTHER PARAMETERS> field in this opcode indicates to a data acquisition device having bar code scanning capabilities which bar code symbologies are enabled or disabled.

Autodiscrimination of Data Transmission Parameters

An additional feature of the preferred embodiment system implements an autodiscrimination function which allows the system to automatically determine the transmission parameters of the system. This feature of the invention will be described in an exemplary manner in conjunction with a host device which communicates with an interface cable **30** in the RS-232 standard format.

To initiate the autodiscriminate function, the user scans a predetermined "autodiscriminate" bar code which is sent to the cable **30** as ADF data. The cable **30** interprets this as an autodiscriminate command an enters a "listen" mode. The user then presses a sequence of keys on the keyboard of the host device, namely the keys "UTU" (selected to appropriately toggle the data lines). The "UTU" data sequence is sent to the cable **30**, which observes the operation of the data lines in order to automatically determine the pertinent RS-232 parameters; namely, baud rate, parity, number of data bits, stop bits, the transmit pin, and the transmission direction. The cable **30** stores these autodisciminated parameters in a local non-volatile memory and uses these parameters to configure subsequent communications with the host device **40**.

The system **10** of the present invention may advantageously be used for bidirectional control of either the host device **40** or the data acquisition device **20**. In addition to requesting and accepting scale data from the data acquisition device **20** and accepting acquired (i.e. decoded) data, the host device **40** can be used to enter control commands to be sent to the data acquisition device **20**. This includes the use of the host's keyboard **47** to enter data acquisition device parameters which are stored in the memory **26**. In addition, host operating parameters can be entered via the host keyboard **47** and sent to the data acquisition device **20** for storage in memory **26**.

In addition, the data acquisition device **20** can be used to control or configure the host device **40**. The user either enters via the keypad **23** or, if applicable, by scanning a special bar code, the required control command/data for transmission to the host device **40**.

FIG. **5** illustrates an alternative embodiment of the invention in which dedicated host formatting modules **51**, **52** and **53** are included in a data acquisition device **20** along with the universal data formatting means **27**. Acquired data is input to each module **51**, **52**, **53** and **27**, and the output of one is selected via a multiplexer **50** for transmission by the transmit/receive buffer **28**. When any of the dedicated host interface modules **51**, **52** or **53** is used, the connector **12** is connected directly to the host device. When the device **20** is used with a host device **40** not compatible with any of the available host interface modules **51**, **52**, or **53**, and the universal data formatting means **27** is used to provided SDCI interface, then the connector **12** is inserted into a host interface cable **30**, for translation as set forth above.

We claim:

1. In a bar code scanning system including a bar code scanner for providing digital signals indicative of scanned data for transmission to a host device, said host device having host-specific input/output data format requirements, said bar code scanning system comprising:

(a) universal data exchange means located in the bar code scanner for providing said digital signals in a universal data exchange format independent of the data format requirements of the host device; and

(b) a host interface module externally coupled between said bar code scanner and said host device, for transmitting acquired digital signals from said bar code scanner to said host device, said interface module comprising means for translating said digital signals from said universal data exchange format to said host-specific input/output data format requirements.

2. The interface system of claim **1** wherein said host interface module comprises a buffer for storing a host parameter data word used to configure communications with the host device; and wherein said universal data exchange means comprises:

a memory for storing a plurality of host parameter data words, each of said host parameter data words being associated with one of a plurality of different host interface modules; and

means for reading from said memory and transmitting to said host interface module the host parameter data word associated with the particular host interface module operatively connected thereto.

3. The interface system of claim **2** wherein:

said host interface module is initially provided with a default mode of communication with said host device;

said host interface module overwrites said default mode when a host parameter data word is provided by said universal data exchange means and operates in accordance with said host parameter data word; and

said interface module operates in accordance with said default mode when a host parameter data word is not provided by said universal data exchange means.

4. The interface system of claim 3 wherein said host interface module and said universal data exchange means execute an initialization process wherein:

said universal data exchange means requests an identification code from said host interface module,

said host interface module provides in response thereto an identification code stored therein,

said universal data exchange means utilizes said provided identification code to read from said memory the host parameter data word associated with said identification code,

said universal data exchange means transmits to said host interface module said host parameter data word read from said memory, and

said host interface module overwrites said default mode with said host parameter data word provided by said universal data exchange means and operates in accordance with said host parameter data word.

5. The bar code scanning system of claim 2 wherein said bar code scanner comprises means for entering host parameter data words by a user.

6. The system of claim 1 adapted to be selectively operated in a normal operating power mode and in a reduced operating power mode, said system further comprising:

timeout determination means located in said bar code scanner for detecting when no data has been acquired within a predetermined timeout period;

means for placing said bar code scanner in said reduced operating power mode in response to said timeout determination; and wherein

said universal data exchange means requests the host interface module to enter into a reduced operating power mode in response to said timeout determination, and said universal data exchange means requests the host interface module to enter into a normal operating power mode in response to a determination that data has been acquired.

7. The interface system of claim 1 wherein said universal data exchange means transmits a poll request in order to determine if a host interface module is connected thereto.

8. The system of claim 1 wherein said bar code scanner transmits to said host device, via said host interface module, control data for controlling operation of said host device.

9. The interface system of claim 1 wherein said host interface module transmits digital data from said host device to said bar code scanner, said interface module further comprising means to translate digital data from said host-specific input/output format to said universal data exchange format.

10. The interface system of claim 9 wherein said host device transmits to said bar code scanner, via said host interface module, control data for controlling the operation of said bar code scanner.

11. The interface system of claim 9 wherein said bar code scanner comprises a memory for storing a plurality of device operating parameter data words, said host device comprises a temporary buffer, and wherein said host device and said universal data exchange means execute the transfer of said device operating parameter data words from said memory to said buffer wherein:

said host device requests in said host-specific input/output format the transfer of said data words;

said host interface module translates said transfer request to said universal data exchange format and transmits said translated request to said universal data exchange means;

said universal data exchange means transfers in response thereto the requested data words to said host interface module;

said host interface module translates said data words to said host-specific input/output format and transmits said translated response to said host device for storage in said buffer.

12. The interface system of claim 11 wherein said host device and said universal data exchange means execute the transfer of said device operating parameter data words from said buffer to the memory of a different bar code scanner wherein:

said host device transmits said data words in said host-specific input/output format;

said host interface module translates said data words to said universal data exchange format and transmits said translated data words to said universal data exchange means; and

said universal data exchange means stores in memory the transmitted data words.

13. The system of claim 1 wherein said universal data exchange format comprises data packets, each data packet having a format comprising:

(a) a protocol and length field indicative of (i) whether the information content of the packet is data or control and status and (ii) the length of the data packet;

(b) an opcode field indicative of the particular message being sent by the packet;

(c) a message source field indicative of the source of the message;

(d) a status field indicative of a retransmit condition, a continue condition, or a low battery power condition;

(e) a data and other parameters field indicative of the data sent in accordance with the particular opcode being implemented; and

(f) a checksum field.

14. The system of claim 1 wherein the host interface module, on command from the bar code scanner, monitors data transmitted by the host device in order to determine said host-specific input/output data format requirements.

15. A host interface module externally coupled between a bar code scanner and a host device and for transmitting acquired digital signals from said bar code scanner to said host device, said interface module comprising:

(a) means for receiving digital signals in a universal data exchange format from the bar code scanner;

(b) means for translating said received digital signals from said universal data exchange format to a host-specific input/output data format; and

(c) means for transmitting said translated digital signals to said host device.

16. The host interface module of claim 15 further comprising:

(d) a buffer for storing a host parameter data word used in communications with the host device; and

(e) means for requesting from said bar code scanner a host parameter data word for storage in said buffer.

17. The host interface module of claim 16 wherein:

said host interface module is initially provided with a default mode of communication with said host device;

said host interface module overwrites said default mode when a host parameter data word is provided by said bar code scanner and operates in accordance with said host parameter data word; and

said host interface module operates in accordance with said default mode when a host parameter data word is not provided by said bar code scanner.

18. The host interface module of claim 15 wherein said universal data exchange format comprises data packets, each data packet having a format comprising:

(a) a protocol and length field indicative of (i) whether the information content of the packet is data or control and status and (ii) the length of the data packet;

(b) an opcode field indicative of the particular message being sent by the packet;

(c) a message source field indicative of the source of the message;

(d) a status field indicative of a retransmit condition, a continue condition, or a low battery power condition;

(e) a data and other parameters field indicative of the data sent in accordance with the particular opcode being implemented; and

(f) a checksum field.

19. A bar code scanner for providing digital signals indicative of acquired data for transmission to a host device comprising:

(a) means for acquiring digital data by a user; and

(b) universal data exchange means for providing said digital signals in a universal data exchange format independent of host-specific input/output data format requirements of the host device, said universal data exchange means being operably connectable to a host interface module device externally coupled between said bar code scanner and said host device, said host interface module device capable of translating said digital signals from said universal data exchange format to a host-specific input/output format.

20. The bar code scanner of claim 19 wherein said universal data exchange means comprises:

a memory for storing a plurality of host parameter data words, each of said host parameter data words being associated with one of a plurality of different host interface modules; and

means for reading from said memory and transmitting to a host interface module the host parameter data word associated with the particular host interface module operatively connected thereto.

21. The bar code scanner of claim 20 further comprising means for entering host parameter data words by a user.

22. The bar code scanner of claim 19 wherein said universal data exchange format comprises data packets, each data packet having a format comprising:

(a) a protocol and length field indicative of (i) whether the information content of the packet is data or control and status and (ii) the length of the data packet;

(b) an opcode field indicative of the particular message being sent by the packet;

(c) a message source field indicative of the source of the message;

(d) a status field indicative of a retransmit condition, a continue condition, or a low battery power condition;

(e) a data and other parameters field indicative of the data sent in accordance with the particular opcode being implemented; and

(f) a checksum field.

23. A method of providing digital signals from a bar code scanner having a universal input/output format to a host device having a host-specific input/output format comprising the steps of:

(a) acquiring in said bar code scanner the digital signals to be provided to said host device;

(b) formatting said digital signals in a universal, no-host-specific input/output format;

(c) transmitting said universally formatted digital signals to a host interface module externally coupled between said bar code scanner and said host device;

(d) translating at said host interface module the universally formatted digital signals to a host-specific format data; and

(e) transmitting said translated digital signals to said host device.

* * * * *

US005854945A

United States Patent [19]

Criscito et al.

[11] **Patent Number: 5,854,945**

[45] **Date of Patent: Dec. 29, 1998**

[54] **BAR CODE SCANNER WITH KEYBOARD SIMULATION**

[75] Inventors: **Herbert V. Criscito**, Denville; **Xin Zheng**, Northvale; **Brian Zhang**, Riveredge, all of N.J.

[73] Assignee: **Opticon, Inc.**, Orangeburg, N.Y.

[21] Appl. No.: **781,345**

[22] Filed: **Jan. 21, 1997**

[51] **Int. Cl.**^{6} **G06F 13/00**

[52] **U.S. Cl.** **395/882**; 395/856; 395/892; 235/435

[58] **Field of Search** 235/435; 705/407; 395/825–827, 832, 840, 856, 882, 892

[56] **References Cited**

U.S. PATENT DOCUMENTS

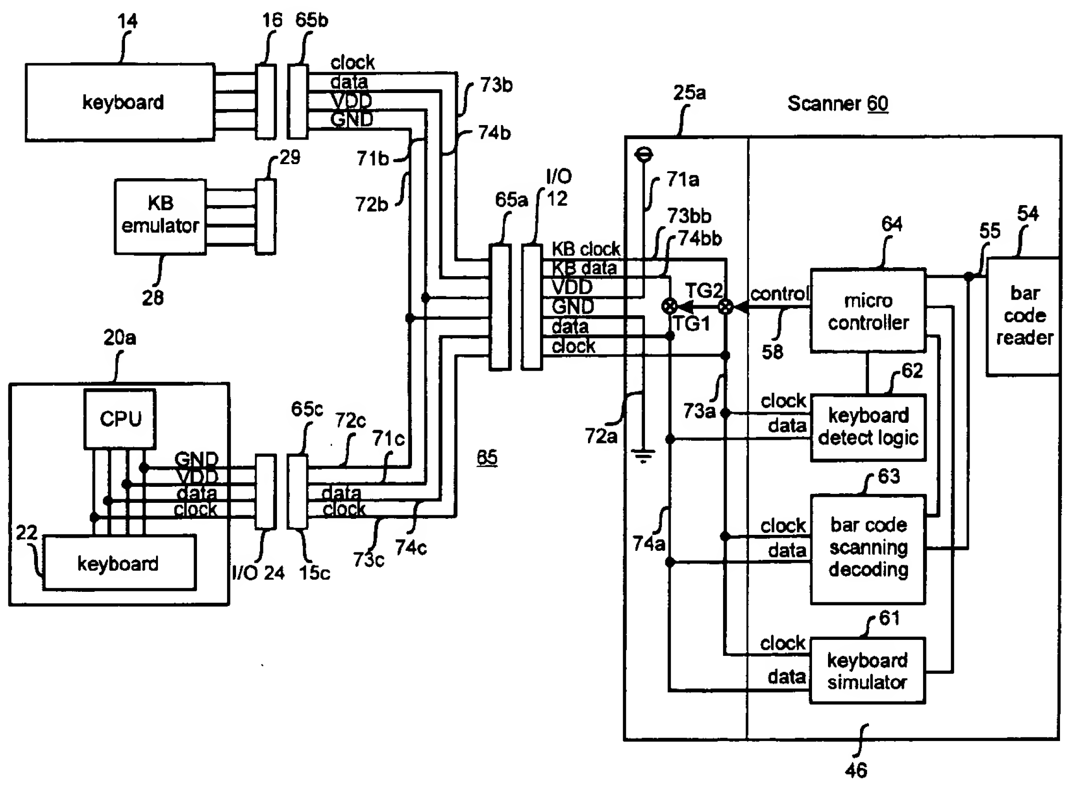
4,642,761	2/1987	Yanagiuchi et al.	395/856
5,034,598	7/1991	Poland	235/435
5,530,893	6/1996	Sugi	395/825
5,615,120	3/1997	Schwartz et al.	705/407

Primary Examiner—Christopher B. Shin
Attorney, Agent, or Firm—Henry I. Schanzer

[57] **ABSTRACT**

A bar code scanner having means for reading bar code data and for processing the data includes means for supplying the processed data to a computer having an input/output (I/O) port designed for the connection thereto of an external keyboard. The scanner includes "communication" means for enabling the scanner either to be "wedged" between an external keyboard and the I/O port or to be directly connected to the I/O port. The scanner's "communication" means includes means for sensing whether an external keyboard is connected to the computer and includes means responsive to sensing the absence of an external keyboard connection for producing signals similar to those produced by an external keyboard and for supplying these signals to the computer in order to establish and maintain communication between the scanner and the computer. The bar code scanner may also include means for responding to various computer commands and/or keyboard commands.

16 Claims, 6 Drawing Sheets



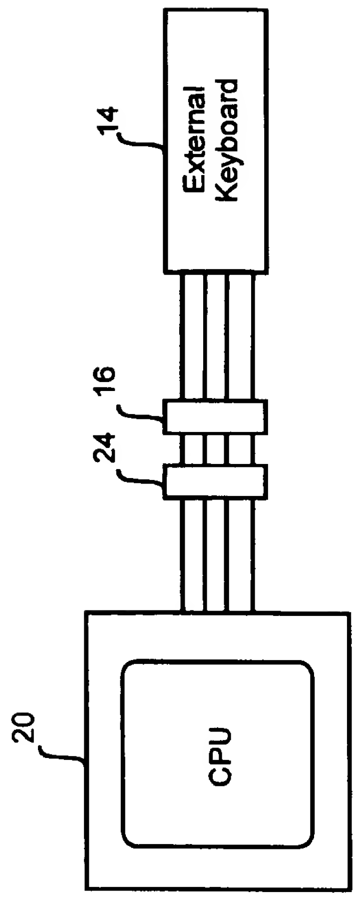


Fig 1 Prior Art

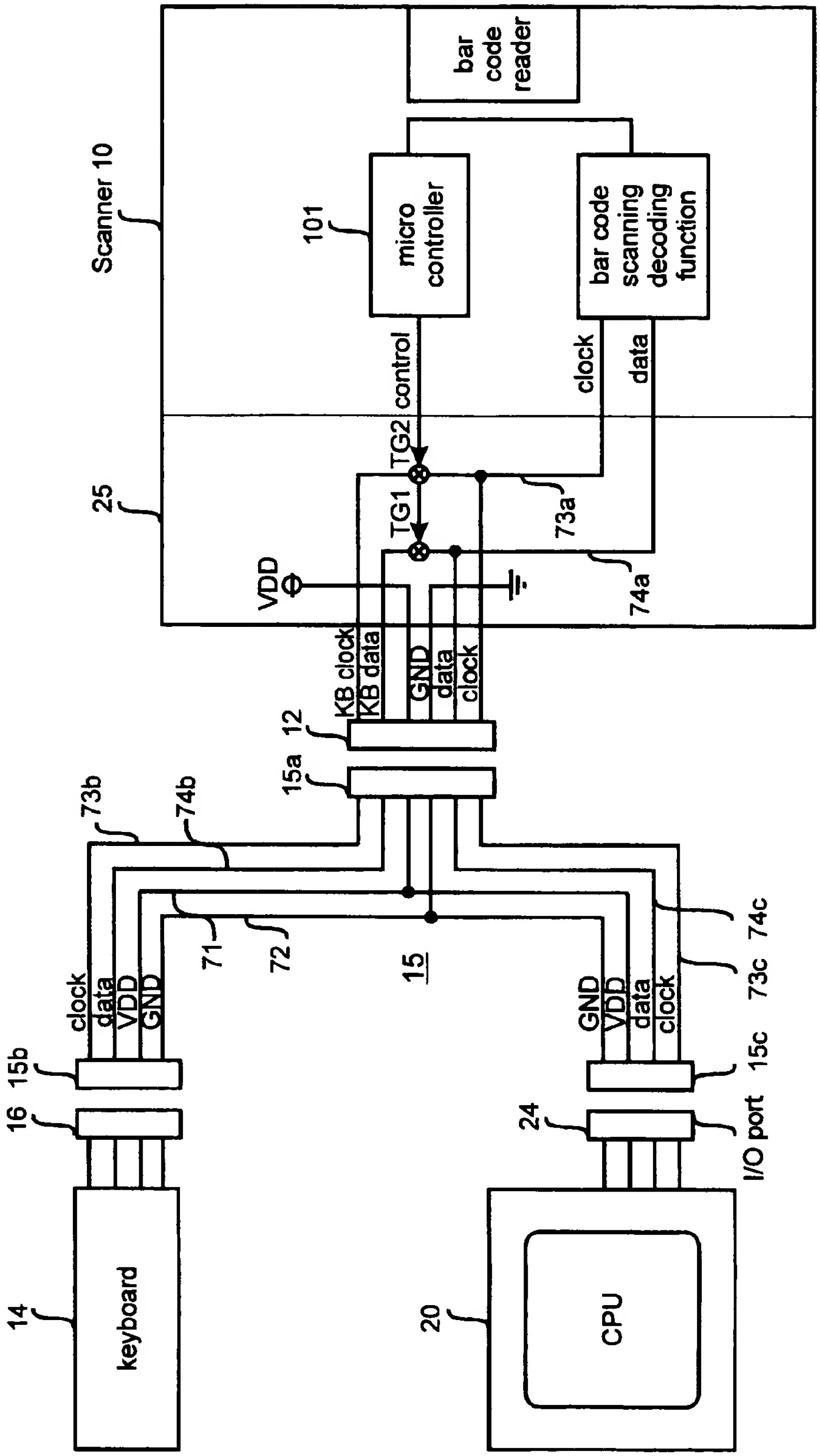


Fig 2 Prior Art

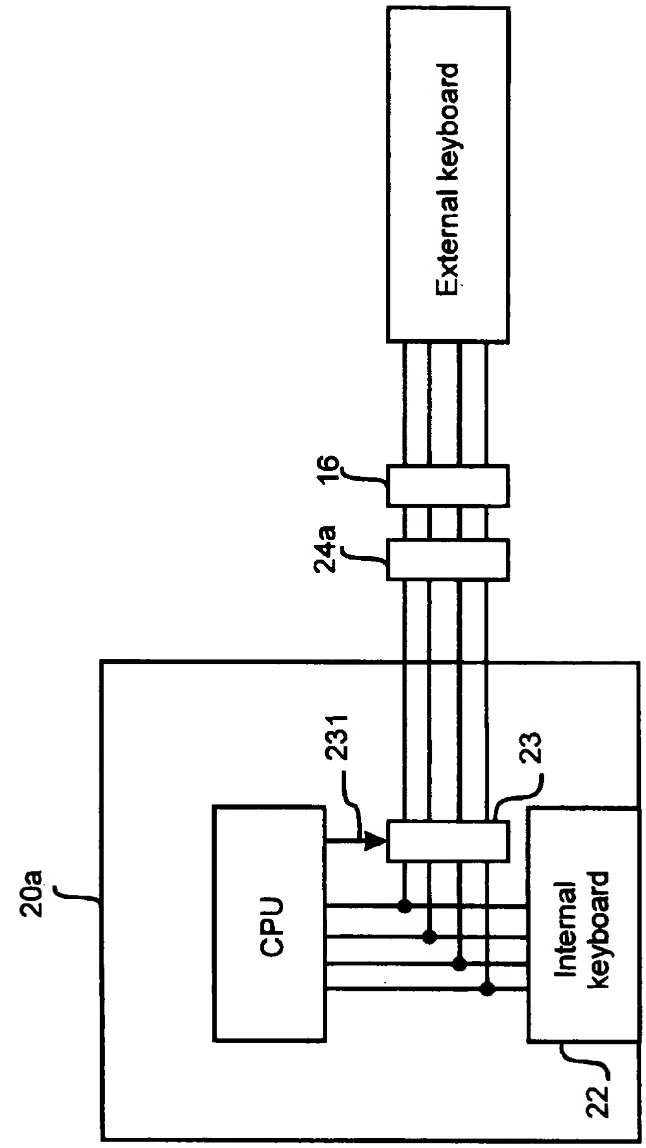


Fig 3 Prior Art

10/29/2003, EAST Version: 1.4.1

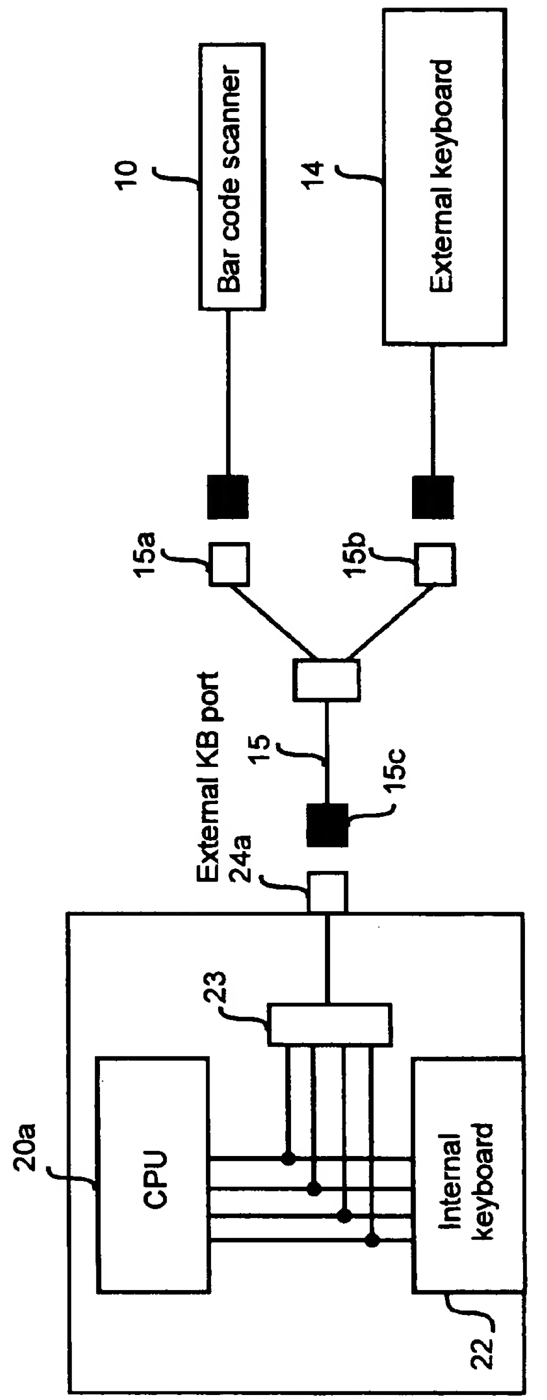


Fig 4 Prior Art

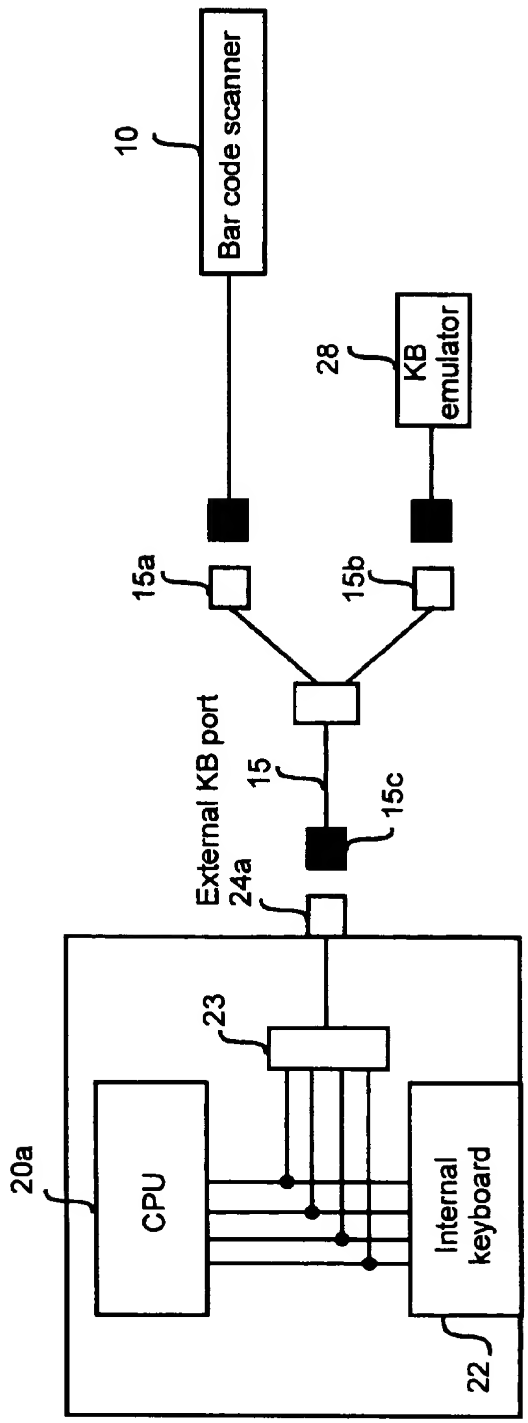


Fig 5 Prior Art

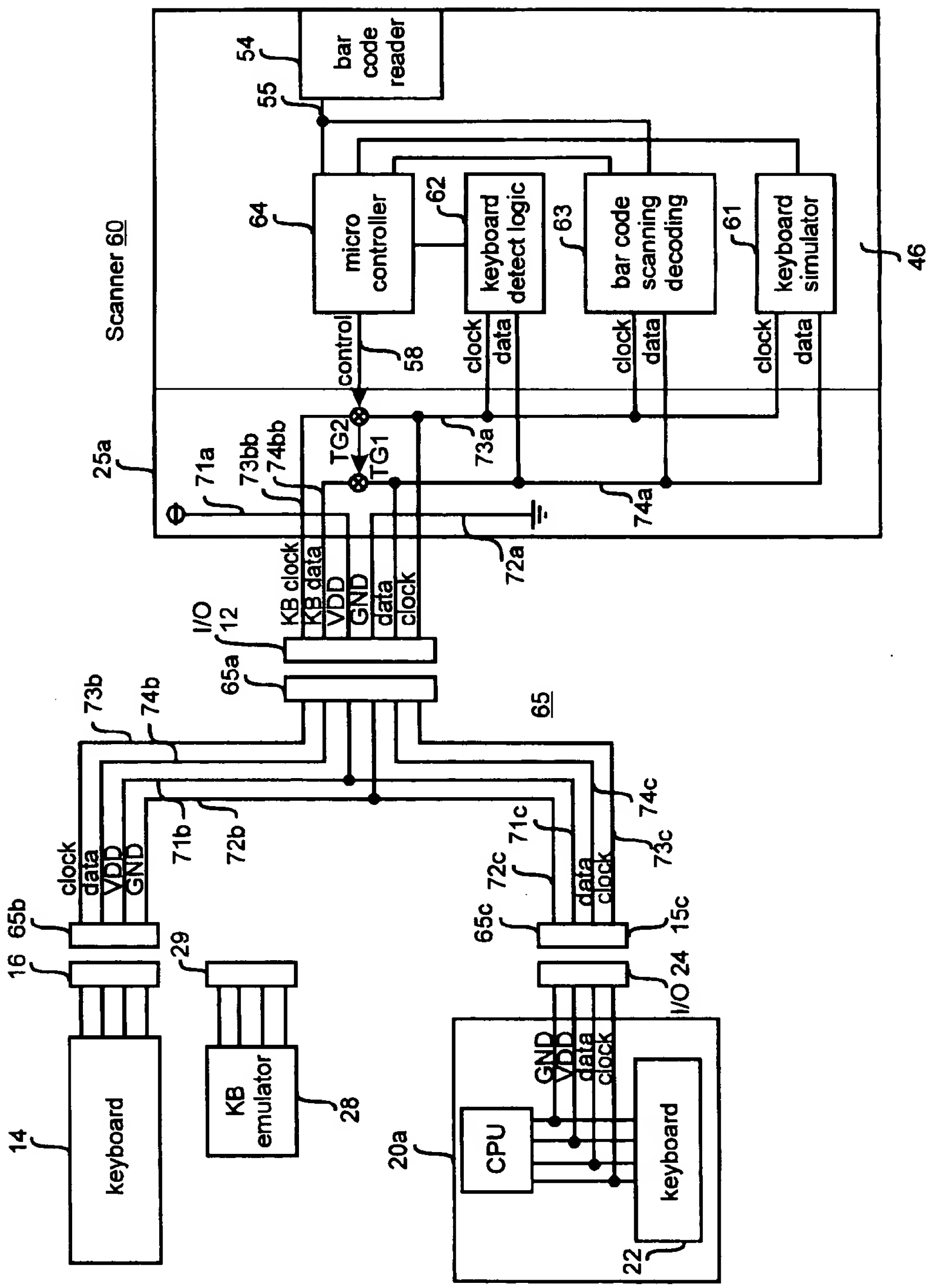


Fig 6

10/29/2003, EAST Version: 1.4.1

BAR CODE SCANNER WITH KEYBOARD SIMULATION

BACKGROUND OF THE INVENTION

This invention relates to bar code scanners and, in particular, to a bar code scanner which includes means for simulating the response of an external keyboard and for rendering the bar code scanner suitable for use with any type of personal computer (PC) including, for example, a desktop or laptop computer.

A bar code scanner is a scanning device which can read, or sense, bar code information and which includes means for converting bar code information into digital information, such as ASCII code, known as "decoded data".

In many applications it is desirable to transfer data read and decoded by a bar code scanner to a personal computer (PC). However, the transfer of decoded data from a bar code scanner to a PC may be problematic. This is so, because a personal computer (PC) does not normally have an input/output (I/O) port designed for the connection thereto of a bar code scanner.

Normally, as shown in FIG. 1, a computer (e.g., CPU **20**) has an external I/O port **24** for the connection thereto of an external keyboard (e.g., KB **14**). A common technique for transferring decoded data from a bar code scanner to a computer, of the type shown in FIG. 1, is to "wedge" a scanner **10** between an external keyboard **14** and a CPU **20** as shown in FIG. 2. In FIG. 2, the bar code scanner **10** has an interface circuit **25** connected or "wedged" between CPU **20** and external keyboard **14**. The keyboard **14** is shown having an I/O port **16** and the scanner **10** is shown having an I/O port **12**. A "Y" cable **15** couples the I/O port **24** of CPU **20** and the I/O port **16** of external keyboard **14** to I/O port **12** of scanner **10**. The Y cable functions to: (a) couple the power lines from CPU **20** to the power lines of the scanner **10** and keyboard **14**; (b) couple the data and clock lines of CPU **20** to the data and clock lines of scanner **10**; and (c) couple the data and clock lines of CPU **20** and scanner **10** via transmission gates TG1 and TG2 of interface circuit **25** to the respective data and clock lines of keyboard **14**. The turn-on and turn-off of transmission gates TG1 and TG2 is controlled by means of circuitry contained within a scanner microcontroller **101**.

In FIG. 2, cable **15** couples CPU **20** via interface circuit **25** to scanner **10** and external KB **14**. The scanner **10** and/or the interface circuit **25**, when coupled via cable **15** to a CPU and a KB, may be referred to as a "wedge" since it is connected or "wedged" between a keyboard (e.g., KB **14**) and a CPU (e.g., **20**).

A "wedge" interface is very desirable because users can easily install a "Y" cable having one end (**15***c*) of the Y cable connected to I/O port **24** with the remaining two ends (**15***a*, **15***b*) of the Y cable being respectively connected to an I/O port **12** of a scanner **10** and an I/O port **16** of a KB **14**.

The bar code scanner **10** controls the communication path between the CPU **20** and the KB **14** by controlling the turn-on and turn-off of TG1 and TG2 [also referred to herein as the "traffic control switch" (TCS)] in interface circuit **25**. When the scanner **10** is powered or when it has no decoded data to send to the CPU, the scanner enables (closes) transmission gates TG1 and TG2 , so the CPU **20** and the KB **14** can communicate directly. When the scanner has decoded data in its data buffer that it is ready to send to the CPU, it disables or turns off (opens) TG1 and TG2 and transmits its decoded data to the CPU **20**. When TG1 and TG2 are disabled, communication between the KB and the CPU is cut-off. Upon completing the transmission of the decoded data, the scanner **10** turns on (enables) TG1 and TG2.

Communication between a computer **20** and an external KB **14** may be conducted directly via 4 lines connected between the computer's I/O port **24** and an I/O port (e.g., **16**) of an external KB **14** as shown in FIG. **1**; where the 4 lines are used for carrying the operating voltages, the clock and the data. Alternatively, communication between a computer and an external KB may be conducted via a wedge, as shown in FIG. **2**.

The CPU and the KB can communicate bi-directionally over the clock and data lines. The source of each of these lines is an open-collector (or drain) device (not shown) in each one of the KB and the CPU which allows either the KB or the CPU to force a line to an inactive (low) level. When there is no communication between the CPU and the KB, both clock and data lines are at an active (high) level. When the CPU sends data to the KB, it forces the DATA line to an inactive (low) level and allows the clock line to go to an active high level. The clock line provides the clocking signals used to clock serial data into and out of the CPU. The KB generates the clock signal for the data transfered between the CPU and the KB. The CPU can prevent the KB from sending data by forcing the clock line to an inactive (low) level.

In the prior art system of FIG. **2**, communication between the CPU and the scanner is one way communication. That is, the CPU supplies commands/data to the scanner **10**, the scanner **10** does not respond to commands from the CPU, it only supplies decoded data to the CPU. When the scanner supplies data to a CPU, just prior to the transmission of decoded data to the CPU, the scanner **10** checks the clock line to determine that it is in an active (high) state (i.e., indicative that the CPU is ready to receive data) at the beginning of a scan code transmission. Once the scanner begins to transmit data it does not again check the status of the line, even when CPU stops receiving data. This may result in the loss of data since the scanner continues to transmit data to the CPU, but the CPU does not accept or process this data.

In the prior art system, if the CPU indicates that there is an error (e.g., parity error) in the data it has received and requests the scanner to "resend" the data, the scanner does not recognize the command and does not check and resend to correct a "receiving" error sensed by the CPU.

In the prior art configuration, bi-directional communication between the computer **20** and the external KB **14** is taken care of by the external KB. The existence of the KB **14** is essential to have the computer **20** keep the I/O port **24** active; which I/O port is effectively designed for interfacing with the external KB.

However, as is well known, there are computers which have built-in internal keyboards. This is the case for most laptop computers and PC based point of sale (POS) systems. As shown, for example, in FIG. **3**, in a computer **20***a* with a built-in keyboards **22** the interconnections between internal keyboard **22** and the processing circuitry of the CPU are "hard wired". As a result, there is not a physical wire connection between the CPU of computer **20***a* and internal KB **22** which can be easily detached. As a result, a bar code scanner can not be easily "Wedged" in between an internal KB **22** and the CPU of computer **20**. Nevertheless, a computer **20***a* with an internal KB **22** usually has an external I/O port **24***a* to enable users to use an alternative external KB **14** as shown in FIG. **3**. In FIG. **3**, the external I/O port **24***a* is shown to be coupled to the CPU and the internal KB via

a control gate (or network) **23**. The CPU may disable control gate **23** (via line **231**) if the CPU senses that an external KB is not connected to I/O port **24***a*.

Where CPU **20***a* has an external I/O port **24***a*, a "Y" cable **15** may be used to "wedge" a bar code scanner **10** between I/O port **24***a* and an external KB **14**, as shown in FIG. **4**. However, as noted above, I/O port **24***a* may be disabled (deactivated) when the CPU senses that an external KB is not connected to I/O port **24***a*. If, and when, that occurs, a scanner, although connected to I/O port **24***a*, could not communicate directly with the computer. In the absence of an external KB, a KB emulator could be connected via cable **15** to the computer and the scanner, as shown in FIG. **5**. However this requires the use of an additional piece of equipment (i.e., external KB emulator **28**) which is undesirable because of the additional cost and the need to use a piece of otherwise unnecessary equipment.

Therefore, it is an object of this invention to enable a bar code scanner to communicate directly with any computer (i.e., one having an internal and/or an external keyboard) regardless of the presence or absence of an external keyboard.

It is a further object of this invention to enable a bar code scanner to be "wedged" between any computer and an external keyboard connection and to communicate bi-directionally with the computer in essentially the same manner as an external keyboard connected in circuit with the computer.

It is a still further object of this invention to resolve various problems of the prior art systems described above.

SUMMARY OF THE INVENTION

A bar code scanner embodying the invention includes means for coupling the scanner to a computer unit (CPU), means for sensing whether an external KB (or a keyboard emulator) is connected to the CPU and means for producing signals simulating the presence of an external keyboard. If the bar code scanner senses that an external keyboard is not present, the bar code scanner produces signals simulating the presence of an external keyboard and supplies those signals to a computer unit to establish and maintain bi-directional communication between the CPU and the scanner.

In one embodiment of the invention, a bar code scanner includes an interface circuit for "wedging" the scanner between an external keyboard and a CPU for controlling the transmission of data and clock signals between the external keyboard and the CPU. The bar code scanner includes means for sensing whether an external keyboard is connected to the CPU, means for producing signals simulating the signals produced by an external keyboard, and means for supplying these "simulated" signals to the CPU when an external keyboard is not connected to the CPU.

In another embodiment of the invention, the scanner includes means for simulating the signals produced by an external keyboard, means for coupling the scanner to an external I/O port of a computer, and means for sensing whether an external keyboard is connected to the computer. If the scanner senses the presence of an external KB, the scanner turns off or disables its KB simulator function. If the scanner does not sense the presence of an external KB, the scanner turns on, or enables, its KB simulator function.

In still another embodiment of the invention, the bar code scanner includes means for sensing the setting or status of certain external KB commands such as "Caps Lock", and means for transmitting to the CPU decoded bar code data consistent with a "Caps Lock" keyboard setting.

In still another embodiment of the invention, the scanner is programmed to receive CPU commands and to check the data and clock lines to conduct bi-directional communication with the CPU, continuously. The scanner can therefore respond to CPU commands while the scanner is sending decoded data to the CPU. A scanner embodying the invention may also include means for sensing the timing and the length of the start up period of any KB connected to the computer and for adjusting the timing of the scanner, whereby users don't need to set up the KB to certain status conditions before a scanning operation.

A bar code scanner embodying the invention may be directly plugged into a computer's external I/O port, even where the I/O port is primarily designed for the connection thereto of an external keyboard. The scanner makes the CPU believe that there is an external KB connected to its I/O port so the computer keeps its I/O port active. The computer's I/O port can then receive input data from the scanner as it does from an external KB. A bar code scanner embodying the invention can thus be "substituted" for an external KB or it can be "wedged" between an external KB (or a KB emulator) to the I/O port of a computer.

Connecting a scanner embodying the invention to an I/O port of a computer increases the decoded data transmission reliability since the scanner responds to commands from the CPU while transmitting decoded data to the CPU.

BRIEF DESCRIPTION OF THE DRAWING

In the accompanying Figures like reference characters denote like components; and

FIG. **1** is a simplified block diagram of a prior art system in which an external keyboard is connected to a computer;

FIG. **2** is a block diagram of prior art system in which a bar code scanner is "wedged" between an external keyboard and a computer;

FIG. **3** is a simplified block diagram of a prior art system in which a computer with an internal keyboard (KB) has an external I/O port for the connection thereto of an external KB;

FIG. **4** is a simplified block diagram of a prior art system in which a computer with an internal KB is coupled via a "Y" cable to an external KB and a bar code scanner;

FIG. **5** is a simplified block diagram of a prior art system in which a computer with an internal KB is coupled via a "Y" cable to a KB emulator and a bar code scanner; and

FIG. **6** is a block diagram of a scanner system embodying the invention.

DETAILED DESCRIPTION OF THE INVENTION

FIG. **6** depicts a bar code scanner **60** and an external keyboard **14** (or a keyboard emulator **28**) which can be coupled to a computer **20***a* via a "Y" cable **65**. The bar code scanner **60** is effectively coupled or "wedged" between the CPU **20***a* and the KB **14** via cable **65**. The computer **20***a*, has an internal keyboard **22** and an I/O port **24**. However, computer **20***a* may be any suitable computer or central processor known in the art, with or without an internal keyboard. In FIG. **6**, as in the prior art, computer **20***a* has 4 lines coming into I/O port **24**; two lines for carrying the clock signals and the data signals and two lines for carrying the operating voltages (VDD and Ground). The line (or lines) for carrying the positive operating voltage (e.g., VDD assumed to be +5 volts) among the CPU, the KB and the scanner is denoted with the prefix **71**. The line(s) for

carrying the negative operating voltage (e.g., ground) among the CPU, KB and scanner is denoted with the prefix **72**. The clock (or CLK) line(s) for carrying the clock signals is denoted with the prefix **73**. The data (or DATA) line(s) for carrying the data is denoted with the prefix **74**.

The keyboard **14**, which may be any suitable external keyboard known in the art, has an I/O port **16** for connecting to the keyboard's clock, data and operating power lines.

The scanner **60** has an I/O port **12** to which six (6) internal lines from the scanner are connected; of the six lines, four lines (**71***a*, **72***a*, **73***a*, **74***a*) are for interconnecting the scanner with the CPU **20***a* and two lines (**73**bb, **74**bb) are for interconnecting the scanner with the external keyboard **14**.

The scanner is interconnected with the CPU and the KB **14** via a "Y" shaped cable **65** having one end, connector **65***a*, connected to scanner I/O port **12**, having another end, connector **65***b*, connected to KB I/O port **16** and having a third end, connector **65***c*, connected to CPU I/O port **24**. Cable **65** may be (but need not be) similar to prior art cable **15** shown in FIG. **2**. Cable **65** functions to interconnect power line **71***c* (the VDD line) from computer **20***a* to power line **71***b* of KB **14** and power line **71***a* of scanner **60**. Likewise, cable **65** functions to interconnect power line **72***c* (ground line) from computer **20***a* to power line **72***b* of KB **14** and power line **72***a* of the scanner. The clock line from computer **20***a* is coupled from I/O port **24***a* via connector **65***c* and clock line **73***c* to connector **65***a* and via I/O port **12** to clock line **73***a* of the scanner. In a similar manner, data from the computer is outputted via I/O port **24** onto DATA line **74***c* which is coupled via connector **65***a* and I/O port **12** to DATA line **74***a* of the scanner. Cable **65** also functions to interconnect the clock and data lines (**73***b* and **74***b*, respectively) of KB **14** to the clock and data lines (73bb and 74bb, respectively) of the scanner and CPU **20***a*.

The barcode scanner **60** shown in FIG. **6** includes a main scanner section **46** and an interface section **25***a*. Section **46** is shown to include a KB simulator section **61**, a KB detect logic section **62**, a bar code scanning decoder section **63**, a microcontroller section **64**, and a bar code reader section **54** to read or sense a bar code located externally to the reader. The bar code reader **54** may be any of a multiplicity of bar code readers which are known in the art. The data read or sensed by the bar code reader is supplied (e.g., via a line **55**) to bar code scanning decoder **63** and to microcontroller **64** for further processing and/or transmission. The decoding and processing of the bar code data is similar to that performed in known systems whose teachings are incorporated herein, and need not be detailed.

The interface circuit **25***a* includes two control gates, **TG1** and **TG2**, which are turned on and off by means of controller **64** via control signals on a control line **58**. Control gates **TG1** and **TG2** may be transmission gates **TG1** and **TG2**, which function to control the connection of the clock and data lines of KB **14** to the clock and data lines of the CPU **20** and the scanner **60**. Interface circuit **25***a* may be (but need not be) similar to interface circuit **25** shown in FIG. **2**.

In contrast to the prior art, the KB simulator function **61** and the KB detect logic function **62** of the bar code scanner **60** enable communication to take place bi-directionally between the CPU **20** and the scanner **60** whether or not a KB (or KB emulator) is connected in circuit with the CPU. That is, connector **65***a* of cable **65** may be connected to I/O port **12** of the bar code scanner **60** and connector **65***c* may be connected to I/O port **24***a*, with connector **65***b* being either connected to or disconnected from I/O port **16** of a KB. Alternatively, the bar code scanner connector **12** may be directly plugged into I/O port **24** of CPU **20***a* and the CPU will interface and interact with scanner **60** as if the scanner were a keyboard. Thus, in systems embodying the invention, connector **65***b* may be selectively connected to I/O port **16** (or I/O port **29** of emulator **28**), or it may be left disconnected (open).

The two controllable transmission gates **TG1** and **TG2** in section **25***a* are for selectively coupling clock line **73***c*/**73***a* to the keyboard clock line **73***b*, and for selectively coupling data line **74***c*/**74***a* to keyboard data line **74***b*. The scanner **60** controls the turn-on and turn-off of transmission gates **TG1** and **TG2** via control line **58**. The turn-on and turn-off of **TG1** and **TG2** in turn controls communication between the CPU **20***a* and the external KB **14** (or the external KB emulator **28**).

When the transmission gates **TG1** and **TG2** are turned-off (open), lines **73***c*/**73***a* and **74***c*/**74***a* are decoupled from lines **73***b* and **74***b*, respectively. However, line **73***c* is always connected to line **73***a* and line **74***c* is always connected to line **74***a*, whereby the sensing and decoding circuitry of the scanner **60** is always connected to sense the status of the computer and/or the presence of an external keyboard.

When the transmission gates **TG1** and **TG2** are turned-on (closed), lines **73***c* and **74***c* are coupled to lines **73***b* and **74***b*, respectively, and the KB **14** (or KB emulator **28**) can communicate with the CPU **20***a*.

In systems embodying the invention the scanner is programmed to respond to all commands from the CPU in exactly the same way as the KB normally responds to the CPU commands. Based on the nature of the command or data signal received from the CPU, the scanner will respond as programmed.

The scanner is programmed to sense whether an external KB or a KB emulator is present as detailed below. Sensing can be achieved using hardware or using software. In the circuit of FIG. **6**, the keyboard detector logic **62** is programmed to automatically detect the presence of an external KB (or a KB emulator) as part of "the KB Power-On Routine". When power is first applied to the CPU, power is distributed from the CPU to the KB and the KB begins with a self-checking procedure. Upon the completion of the selfchecking procedure, the KB sends a completion code (Hex AA) to the CPU. Note that when the scanner is "wedged" between the KB and the CPU, direct communication between the KB and the CPU is initially enabled because when the scanner is powered up (with the CPU and the KB connected in circuit), the scanner turns on the "traffic control switch"; i.e., gates **TG1** and **TG2** are initially turned-on to provide a connection between the CPU and the KB. Thus, upon application of power to the system (i.e., the CPU, KB and scanner), the KB goes through a self-checking sequence during a start up period. At the end of the start up period, the KB produces a set and/or sequence of "completion code" signals and supplies the signals over the data and clock lines to the CPU and the scanner. If a KB is not present (i.e., not connected to the CPU) this set of KB signals is not produced. Scanners embodying the invention include means for sensing the set or sequence of "completion code" signals generated by a keyboard.

If the KB detect logic **62** of scanner **60** senses the set of signals indicating that a KB is present it sets a register or a random access memory (RAM) located in microcontroller **64** (or in another control section of the scanner) to a first condition ("KB present") to indicate the presence of a KB. In response to the "KB present" condition the scanner microcontroller **64** is programmed and inhibited from clock-

ing commands and/or data out of the CPU except during "decoded data transmission period". When a "KB present" condition exists, the keyboard is responsible for clocking data from the CPU.

If the scanner does not sense the presence of the set of signals, it sets the register to a second condition ("KB absent") indicative of the absence of a KB. When the scanner is set to the "KB absent" condition, the transmission gates TG1 and TG2 are turned-off and the scanner clocks the data on the data line between the CPU and the scanner. That is, when the scanner is set to the "KB absent" condition, the scanner clocks out data from CPU (i.e., the scanner supplies clock signals to clock out data from the CPU). Thus, if the "KB absent" condition is set, the scanner clocks the data on the data line, and, if the "KB present" condition is set, the scanner does not clock the data on the data line. Note that each time power is applied to the CPU and to a scanner connected to the CPU, the scanner checks for the presence of an external KB (or KB emulator) during a "start up" period, which is a predetermined period of time after power is applied to the CPU. After being powered up, a CPU distributes power to the devices connected to its I/O port **24**. The CPU then waits for a fixed period to receive a signal from the external KB that it is powered-on. If a power on signal typically generated by an external KB, is not detected during this fixed period time, the CPU normally sends a "reset" command to the KB. In the absence of a scanner embodying the invention, if the KB fails to clock the command data from the CPU, the CPU would conclude that there is no evidence connected to its I/O port so that it would set the clock "low" and would subsequently ignore any input from the KB port. At this point, the KB port is technically dead.

The scanner also has a waiting time period referred to herein as a "KB detecting period" which is a period of time during which the scanner waits for a power-on signal from the KB, which power-on signal is normally sent out by a KB after it is powered-up. If the power-on signal is not detected during this "KB detecting period", the scanner of FIG. **6** sends out a simulated KB power-on signal to tell the CPU that there is a KB-like-device (i.e., the scanner) attached to I/O port **24** and the scanner also turns off the TCS. The "KB detecting period" time is normally pre-set at the factory (i.e., during manufacture) when the "KB detecting period" time is longer than the CPU's cut off time, the scanner is programmed to recognize this situation by detecting that the clock line is low for too long a period time. The scanner is programmed to then automatically adjust the internal software switches to reduce the "KB detecting period" and to save the switch settings into an electronically erasable programmable read only memory (EEPROM) in the microcontroller section. When the next time the scanner is powered-up, the new setting of "KB detecting" which is controlled by the software switches saved in the EEPROM function to define the "KB detecting" time period. After the start-up period, the CPU and/or the scanner no longer check for the presence/absence of a KB until the next time power is applied.

A feature of the scanner embodying the invention is that it is programmed to automatically adjust the length of time of the "start-up" period to fit the different timing periods associated with different keyboards (or KB emulator). This is set up each time a scanner is plugged into the system or unplugged from the system.

Where a KB is not present, a CPU and the scanner connected to the CPU communicate bi-directionally with each other.

In response to a CPU generated request-to-send (RTS) signal, after the scanner **60** detects an RTS command from the CPU, the scanner **60** interrupts its current scanning and decoding operation and generates a clock signal to clock out command or data from the CPU into the scanner.

When the scanner **60** has scanned data and the data is ready to be sent to the CPU, the scanner is programmed to sense that the CPU **20** is ready to receive the data by checking whether the CLK line is active (high).

The scanner **60** may be directly connected to I/O **24** of the CPU **20***a*. CPU **20***a* may have an internal keyboard **22** as shown in FIG. **6** and may be a "Laptop" computer. In the laptop mode (or whenever the scanner is directly connected to a computer's I/O port) the scanner acts the same as a keyboard. The scanner will receive commands from PC system at any time after power on. Also, it will respond to commands which it receives from the PC and set or reset corresponding software switches to configure the scanner. Like any keyboard, the scanner can be automatically switched between different types of PC's.

Unlike prior art wedge scanner configuration, there is not another keyboard device attached to the scanner. The scanner in accordance with the invention may be automatically set to detect the presence of a KB. If a keyboard is found, the scanner will be configured as a wedge scanner. Otherwise, the scanner is configured as a Laptop scanner.

The scanner of the invention is fully keyboard compatible. In addition to wedge keyboard program, additional scan code table is added to match with keyboard. Like keyboard, the new scanner is capable of receiving and responding to PC system command. According to the PC command, scanner can be initialized and set or reset specific switches. By communicating with PC terminal, the new scanner becomes a part of PC system which is like an external keyboard device.

When a KB command such as CAPS LOCK is generated (i.e., by depressing the Caps Lock key on the KB), the command is transmitted from the KB to the CPU. The CPU then sends back a signal to the KB to cause the turn-on of the Caps Lock light and to indicate that the command has been received and the CPU will handle the data received from the KB in accordance with the command.

When a scanner is "wedged" between the KB and the CPU, the scanner information sent to the CPU after the CAPS LOCK command will be subjected to unwanted and undesirable modifications.

In the scanner embodying the invention, the scanner senses the "CAPS LOCK signal" sent out by the CPU and the scanner then modifies the information it sends out to the CPU so that the scanner data will not be subjected to undesirable modifications by the CPU.

The scanner embodying the invention checks whether the CPU is sending information at the same time as the scanner. This is referred to herein as "line contention detection". When the CPU sends information it pulls down the CKL line. When the scanner senses this condition, the scanner stops sending information to the CPU as described below. The scanner sends out information (i.e., a scan code) or data formatted to have a length of 11 bits. The first bit is a start bit, followed by 8 data bits, which are then followed by a parity bit and a stop bit. In the prior art, the scanner was not programmed or designed to check the status of the CPU before each bit. In the scanner according to the invention, the scanner is designed and/or programmed to sense the status of the clock before clocking each bit of data to the CPU. If the scanner senses that the CLK line has been pulled down

(i.e., the CPU is not ready to receive data) the scanner stops sending information to the CPU. This is so, except where the scanner has already sent out a parity bit. If the parity bit has been sent out the scanner continues sending the next signal which is the stop bit. If the parity bit has not been sent, the scanner stops sending data immediately and releases the clock and data lines.

In the prior art system of FIG. 1, when decoded data in the scanner's buffer was ready to be transmitted to the CPU, the scanner turned off the TCS (i.e., transmission gates TG1 and TG2). During the TCS off period of time, the communication between the CPU and the KB was totally off, since the scanner was not capable of receiving data from the CPU, the communication between the CPU and the scanner was one way communication. That is, the scanner is only capable of sending data to the CPU. The scanner can not receive data from the CPU because it is not designed to receive this information. One of the problems of this one way communication is lack of error recovery. When the data received by the CPU contains a receiving error, (e.g., a parity error), the CPU requests the KB (the CPU never knows that the scanner has cut off communication between the CPU and the KB) to resend the data. The CPU does this by sending a "RESEND" command to the KB which includes pulling the clock line high and the data line low (request-to-send) (or RTS) and then waiting for the KB to clock out the "RESEND" command from the CPU. The prior art scanner was not capable of detecting the RTS condition and was not capable of clocking out a "RESEND" command from the CPU. Instead the scanner would keep sending the next bits of data which could cause additional communication error between the CPU and the scanner.

In contrast thereto, the scanner **60** of FIG. **6** embodying the invention includes the means (e.g., is programamed) to sense the "RESEND" command (i.e., the RTS command) and to clock out the "RESEND" command from the CPU. After receiving the "RESEND" command, the scanner then resends the full data message previously sent to the CPU.

What is claimed is:

1. In a bar code scanner having means for reading and processing bar code data and including a data line and a clock for either supplying signals to a computer or for receiving signals from the computer, means for interconnecting the scanner and a computer comprising:

an interface circuit including: (a) means for direct current connecting the signal lines of the scanner to the signal lines of a computer; and (b) selectively enabled gating means for coupling the signal lines of an external keyboard to the signal lines of the computer and the scanner;

said scanner including means for enabling said gating means whereby signals produced by an external keyboard, if present, are coupled to said signal lines;

keyboard dectect means in said scanner for sensing whether any signals indicative of the presence of an external keyboard are produced on said signal;

means in said scanner for producing signals similar to those produced by a keyboard; and

means for supplying to the data and clock lines of a computer the scanner generated signals which are similar to those produced by an external keyboard in response to the scanner sensing that a keyboard is not present in order to establish and maintain communication between the scanner and the computer.

2. In a bar code scanner having means for reading bar code data located external to the scanner and for processing the data and having means for supplying the processed data to an input/output (I/O) port of a computer, where the I/O port is designed for, normally, connection thereto an external keyboard, the improvement comprising:

communication means within the scanner for enabling the scanner either to be connected between an external keyboard and the I/O port or to be directly connected to the I/O port, said communication means including means for sensing whether an external keyboard is connected to the computer and means responsive to sensing the absence of the connection of an external keyboard for producing signals similar to those produced by an external keyboard and for supplying these signals to the computer in order to establish and maintain communication between the scanner and the computer.

3. A bar code scanner comprising:

means for reading bar code data and supplying the bar code data to a signal processor;

means for coupling the signal processor to scanner signal lines;

an interface circuit including: (a) a first section for coupling thereto signal lines from a computer; (b) means for direct current connecting the scanner signal lines to the computer signal lines; (c) a second section for coupling thereto signal lines from an external keyboard; (d) selectively enabled gating means coupled between said second section and said computer signal lines for, when said gating means is enabled, connecting the external keyboard signal lines to the computer and scanner signal lines;

means for enabling said gating means whereby signals produced by an external keyboard, if present, are coupled to said computer and scanner signal lines;

keyboard detect means for sensing whether any signals indicative of the presence of an external keyboard are produced on said computer and scanner signal lines;

means for producing signals similar to those produced by an external keyboard and for supplying these similar signals to the computer and scanner signal in response to the scanner sensing that a keyboard is not present in order to establish and maintain communication between the scanner and the computer.

4. A bar code scanner suitable for use with a computer having an input/output (I/O) port designed for the connection thereto of an external keyboard, comprising:

means for coupling said bar code scanner to said I/O port of said computer and means within said bar code scanner for sensing whether an external keyboard is coupled to said I/O port; and

means within said scanner responsive to sensing the absence of an external keyboard for producing signals similar to the signals produced by an external keyboard and for supplying said similar signals to said in order to establish and maintain communication between the scanner and the computer.

5. A bar code scanner as claimed in claim 1, wherein an external keyboard, when connected to a computer, produces a certain signal sequence when power is first applied to the external keyboard; and

wherein said bar code scanner includes means for producing signals similar to said certain signal sequence, and in response to sensing the absence of an external keyboard produces said similar signals for supplying them to the computer.

6. A bar code scanner as claimed in claim 5, wherein said scanner senses a "caps lock" command initiated by an external keyboard and in response thereto modifies the data it supplies to the computer for ensuring that the bar code data it supplies to the computer is processed correctly.

7. A bar code scanner as claimed in claim 5 wherein said means within said scanner for sensing whether an external keyboard is coupled to said I/o port includes means for disabling said means for producing signals similar to said certain signal sequence in response to sensing the presence of an external keyboard.

8. A bar code scanner as claimed in claim 1, wherein said means within said bar code scanner for sensing whether an external keyboard is connected to the computer includes a data line for sensing data information and a clock line for sensing clocking information,;

wherein said computer also includes a data line and a clock line; and

wherein said bar code scanner includes means for coupling its data and clock lines to those of a computer.

9. A bar code scanner as claimed in claim 1, wherein said means for coupling said scanner to said I/O port of said computer includes an interface circuit for enabling selected signal lines of the scanner to be directly connected to selected signal lines of a computer; and wherein the interface circuit includes gating means for the connection thereto of selected signal lines from an external keyboard to be connected to the selected signal lines of the scanner and of a computer.

10. A bar code scanner as claimed in claim 9, wherein said scanner includes means for initially enabling said gating means, and wherein said means for sensing whether and external keyboard is coupled to said I/O port includes means for sensing whether a certain signal sequence is produced on said signal lines within a predetermined period of time after the application of an operating potential by a computer.

11. A bar code scanner as claimed in claim 1, wherein the scanner includes means for sensing command signals produced by the computer and means for responding to these commands.

12. A bar code scanner as claimed in claim 11, wherein said scanner constantly senses the signal lines interconnecting the scanner and the computer for providing continuous error correction.

13. A bar code scanner as claimed in claim 11, wherein said scanner constantly senses the signal lines interconnecting the scanner and the computer for resending data to the computer in response to a resend request from the computer.

14. A bar code scanner as claimed in claim 1 suitable for use with a computer, wherein said computer is a computer having an internal keyboard.

15. A bar code scanner as claimed in claim 1 wherein the scanner includes means for supplying previously read bar code data to a computer and for monitoring the computers readiness to accept the data.

16. A bar code scanner as claimed in claim 15 wherein the scanner includes means for continuously sensing whether the computer is no longer ready to accept data for terminating transmission of bar code data to the computer.

* * * * *

US005848292A

United States Patent [19]

Nathan

[11] **Patent Number: 5,848,292**

[45] **Date of Patent: Dec. 8, 1998**

[54] **SYSTEM FOR DATA TRANSMISSION BETWEEN A WEDGE MICROCONTROLLER AND A PERSONAL COMPUTER MICROCONTROLLER BY DISCONNECTING THE KEYBOARD MICROCONTROLLER AND PLACING THE SAME IN HOLD STATE**

[75] Inventor: **Robert H. Nathan**, Atlanta, Ga.

[73] Assignee: **NCR Corporation**, Dayton, Ohio

[21] Appl. No.: **666,163**

[22] Filed: **Jun. 19, 1996**

[51] **Int. Cl.^{6}** **G06F 13/00**

[52] **U.S. Cl.** **395/822**; 395/858; 395/860; 395/871; 395/893

[58] **Field of Search** 395/728, 821, 395/822, 858, 860, 871, 893; 400/472

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,799,144	1/1989	Parruck et al.	395/822
5,101,481	3/1992	Anger et al.	395/309
5,179,375	1/1993	Dick et al.	340/825.51
5,220,522	6/1993	Wilson et al.	364/709.04
5,258,604	11/1993	Behrens et al.	235/462
5,263,171	11/1993	Asprey	395/728
5,438,186	8/1995	Nair et al.	235/449
5,667,319	9/1997	Satloff	400/472

Primary Examiner—Thomas C. Lee
Assistant Examiner—Ki S. Kim
Attorney, Agent, or Firm—James P. Davidson; Peter H. Priest

[57] **ABSTRACT**

A system for interfacing at least one peripheral device with a personal computer is disclosed as including a personal computer microcontroller, a keyboard microcontroller directly connected to the personal computer microcontroller by means of a clock line and a data line, a switching device positioned between the personal computer microcontroller and the keyboard microcontroller, and a wedge microcontroller to which the peripheral device may be connected. The wedge microcontroller further includes a first set of connections to the clock and data lines positioned between the switching device and the keyboard microcontroller, a second set of connections to the clock and data lines positioned between the switching device and the personal computer microcontroller, and a switch control connection to the switching device for opening and closing the switching device. The direct connection between the personal computer microcontroller and the keyboard microcontroller can be disabled by opening the switching device so that the wedge microcontroller may input data from the peripheral device into the personal computer microcontroller.

17 Claims, 3 Drawing Sheets

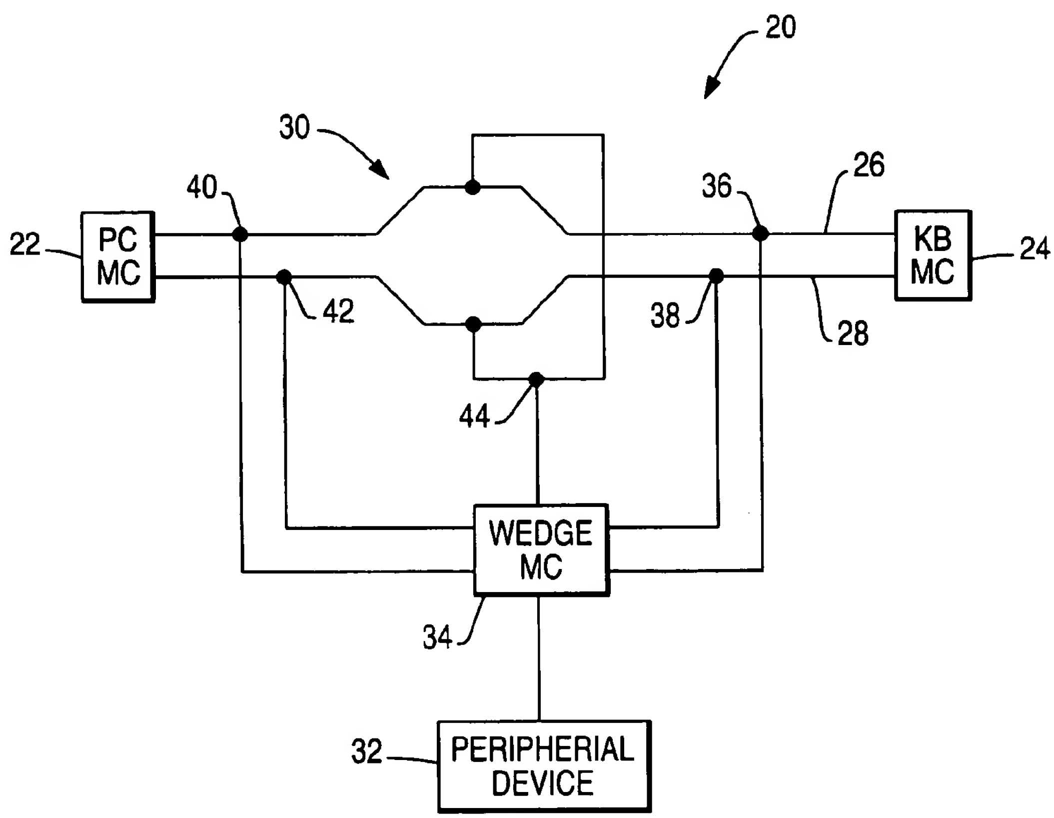


FIG. 1
PRIOR ART

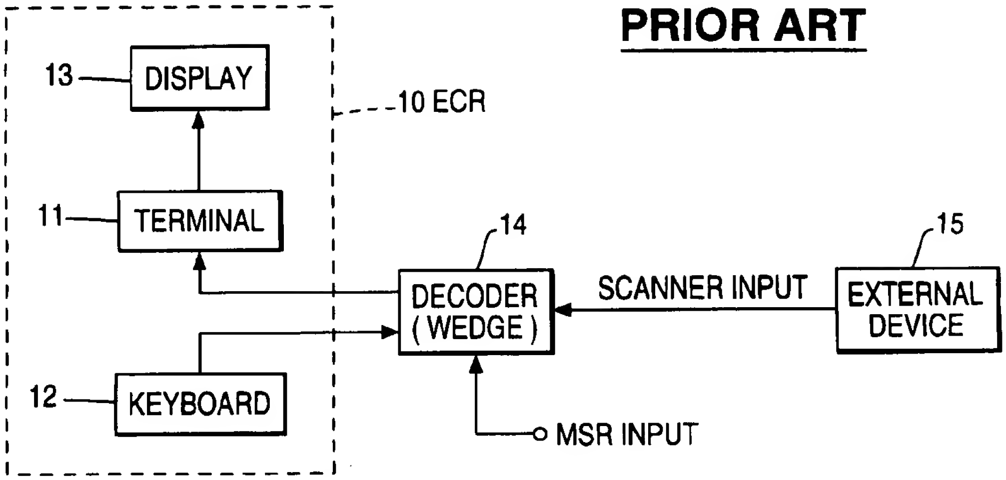


FIG. 2

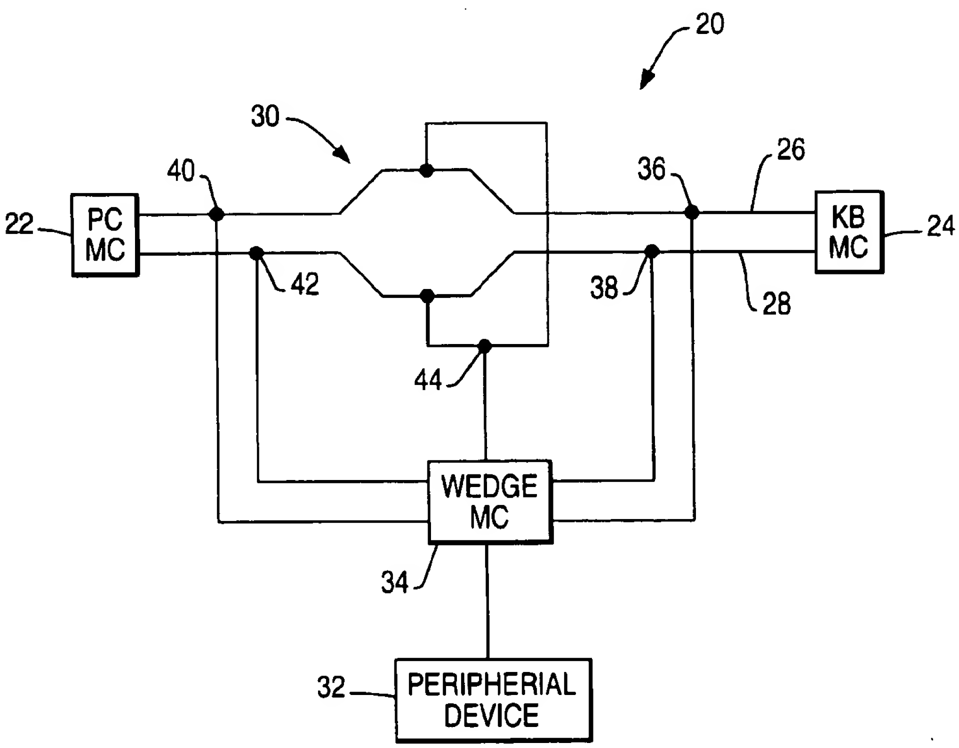


FIG. 3A

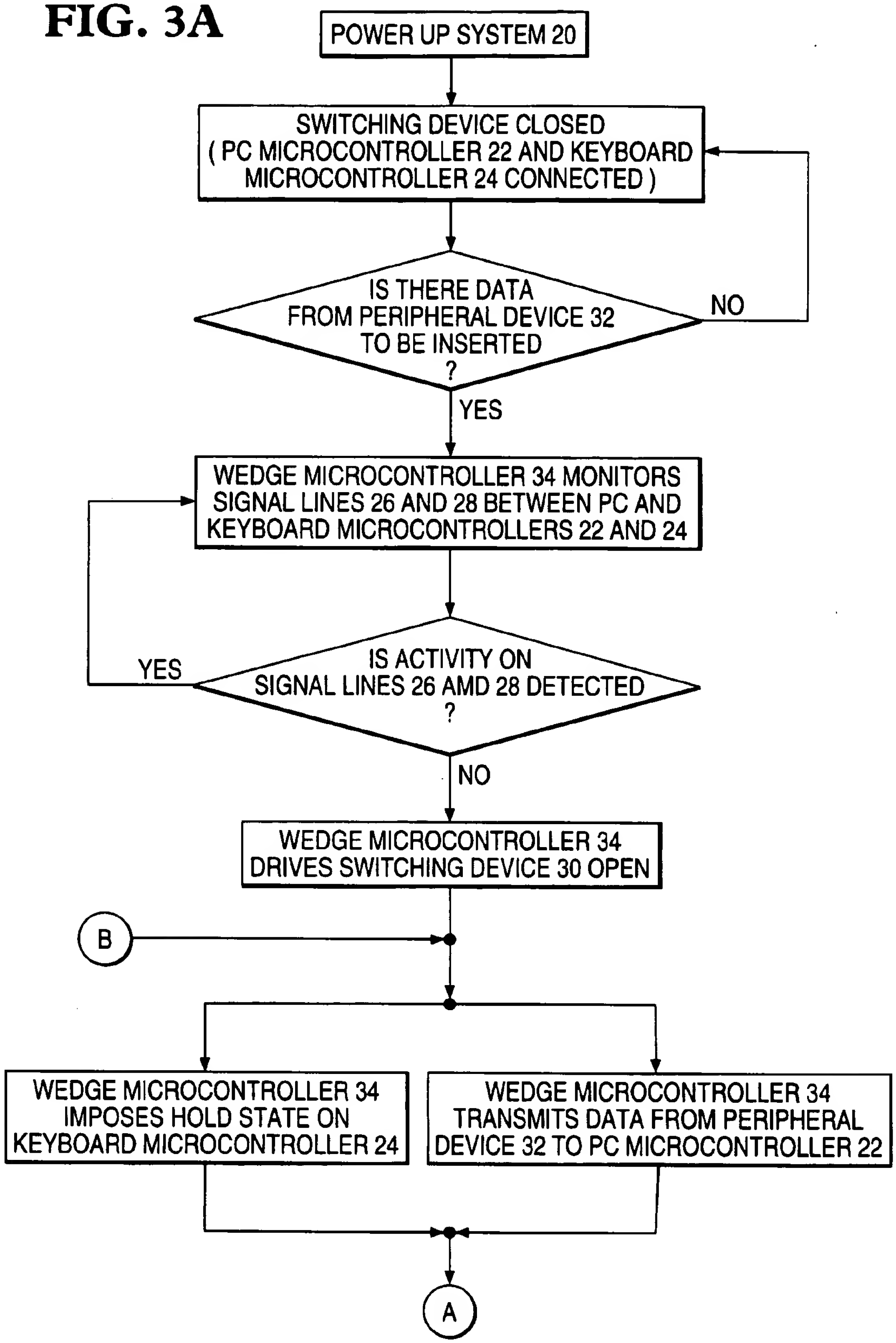
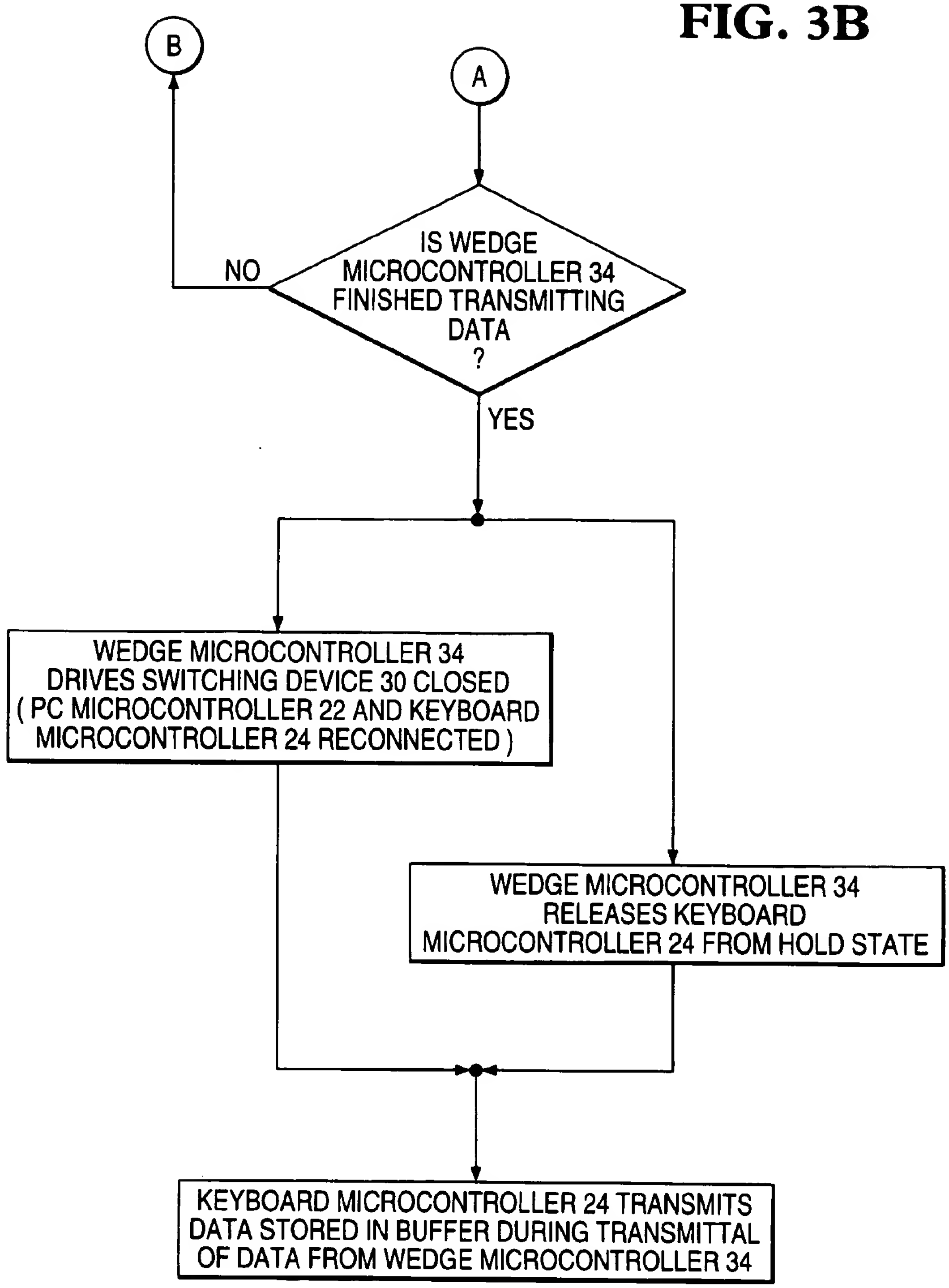

POWER UP SYSTEM 20
SWITCHING DEVICE CLOSED (PC MICROCONTROLLER 22 AND KEYBOARD MICROCONTROLLER 24 CONNECTED)
IS THERE DATA FROM PERIPHERAL DEVICE 32 TO BE INSERTED ?
NO
YES
WEDGE MICROCONTROLLER 34 MONITORS SIGNAL LINES 26 AND 28 BETWEEN PC AND KEYBOARD MICROCONTROLLERS 22 AND 24
IS ACTIVITY ON SIGNAL LINES 26 AMD 28 DETECTED ?
YES
NO
WEDGE MICROCONTROLLER 34 DRIVES SWITCHING DEVICE 30 OPEN
B
WEDGE MICROCONTROLLER 34 IMPOSES HOLD STATE ON KEYBOARD MICROCONTROLLER 24
WEDGE MICROCONTROLLER 34 TRANSMITS DATA FROM PERIPHERAL DEVICE 32 TO PC MICROCONTROLLER 22
A

FIG. 3B


B
A
IS WEDGE MICROCONTROLLER 34 FINISHED TRANSMITTING DATA ?
NO
YES
WEDGE MICROCONTROLLER 34 DRIVES SWITCHING DEVICE 30 CLOSED (PC MICROCONTROLLER 22 AND KEYBOARD MICROCONTROLLER 24 RECONNECTED)
WEDGE MICROCONTROLLER 34 RELEASES KEYBOARD MICROCONTROLLER 24 FROM HOLD STATE
KEYBOARD MICROCONTROLLER 24 TRANSMITS DATA STORED IN BUFFER DURING TRANSMITTAL OF DATA FROM WEDGE MICROCONTROLLER 34

SYSTEM FOR DATA TRANSMISSION BETWEEN A WEDGE MICROCONTROLLER AND A PERSONAL COMPUTER MICROCONTROLLER BY DISCONNECTING THE KEYBOARD MICROCONTROLLER AND PLACING THE SAME IN HOLD STATE

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates generally to wedge interfaces which enable peripheral devices to communicate with a personal computer and, in particular, to a manner of connecting a wedge interface to a personal computer which permits a direct connection between the personal computer and a keyboard.

2. Description of Related Art

It is well known that the standard interface between a personal computer and a keyboard is based on a design which incorporates an electronic circuit having four wires (CLOCK, DATA, +5 volts, and Ground). These four wires are utilized to connect a microcontroller in the keyboard to a microcontroller in the personal computer (or equivalent circuitry). Of course, the transfer of data depends on a well defined signaling convention between the two microcontrollers.

Various peripheral devices have been developed which are inserted or "wedged" in the circuit between the keyboard and personal computer. Examples of such peripheral or wedge devices include magnetic stripe readers, bar code readers, laser guns, and optical character readers. These devices are currently inserted so as to permanently split the connection between the keyboard microcontroller and the personal computer microcontroller by inserting a wedge microcontroller therebetween. An example of such a design is disclosed in U.S. Pat. No. 5,179,375 to Dick et al. The wedge microcontroller connected in this manner therefore serves as a store-and-forward switch, presenting an interface to the keyboard which looks (to the keyboard) like a host personal computer and an interface to the personal computer which looks (to the personal computer) like a keyboard. Accordingly, the keyboard's microcontroller delivers data to the wedge microcontroller when the keyboard has data to send to the personal computer, after which the wedge microcontroller transfers that data to the personal computer microcontroller. When the wedge microcontroller has data to send to the personal computer from a peripheral device, it is able to do so at will.

However, the above-described design has several detriments associated therewith. For example, timing problems occur periodically because the personal computer communicates only indirectly with the keyboard. Additionally, the wedge microcontroller incurs the overhead of becoming a full store-and-forward keystroke message controller, which adds significant initial development time to the wedge microcontroller and exposes the wedge microcontroller to future modifications in keyboard communications stemming from keyboard functionality.

In light of the foregoing, a primary objective of the present invention is to provide a system for interfacing at least one peripheral device with a personal computer which permits direct connection between the microcontroller of the personal computer and the microcontroller of a keyboard.

Another object of the present invention is to provide a system for interfacing at least one peripheral device with a personal computer which minimizes interruption of a direct connection between the personal computer microcontroller and the keyboard microcontroller.

A further object of the present invention is to provide a system for interfacing at least one peripheral device with a personal computer which enables data to be transmitted to the personal computer microcontroller from only one source.

Yet another object of the present invention is to provide a system for interfacing at least one peripheral device with a personal computer in which a microcontroller of the personal computer perceives a wedge microcontroller as a keyboard microcontroller and the keyboard microcontroller perceives the wedge microcontroller as the personal computer microcontroller.

Still another object of the present invention is to provide a system for interfacing at least one peripheral device with a personal computer in which the direct connection between the personal computer microcontroller and the keyboard microcontroller can be dynamically disconnected and reconnected.

Another object of the present invention is to provide a method of periodically connecting a wedge microcontroller to a personal computer microcontroller for data transmission, where the personal computer microcontroller is otherwise directly connected to a keyboard microcontroller.

A still further object of the present invention is to provide a method of transmitting data from a wedge microcontroller to a personal computer microcontroller, where the personal computer microcontroller is directly connected to a keyboard microcontroller.

These objects and other features of the present invention will become more readily apparent upon reference to the following description when taken in conjunction with following drawing.

SUMMARY OF THE INVENTION

In accordance with one aspect of the present invention, a system for interfacing at least one peripheral device with a personal computer is disclosed as including a personal computer microcontroller, a keyboard microcontroller directly connected to the personal computer microcontroller by means of a clock line and a data line, a switching device positioned between the personal computer microcontroller and the keyboard microcontroller, and a wedge microcontroller to which the peripheral device may be connected. The wedge microcontroller further includes a first set of connections to the clock and data lines positioned between the switching device and the keyboard microcontroller, a second set of connections to the clock and data lines positioned between the switching device and the personal computer microcontroller, and a switch control connection to the switching device for opening and closing the switching device. In this way, the direct connection between the personal computer microcontroller and the keyboard microcontroller can be disabled by opening the switching device so that the wedge microcontroller may input data from the peripheral device into the personal computer microcontroller.

In accordance with a second aspect of the present invention, a method of periodically connecting a wedge microcontroller to a personal computer microcontroller for data transmission is disclosed, wherein the personal computer microcontroller is directly connected to a keyboard microcontroller by means of a clock line and a data line. The method includes the steps of: providing a switching device

in the clock and data lines between the personal computer microcontroller and the keyboard microcontroller; providing the wedge microcontroller with a first set of connections to the clock and data lines at a location between the switching device and the keyboard microcontroller; providing the wedge microcontroller with a second set of connections to the clock and data lines at a location between the switching device and the personal computer microcontroller; and, permitting the wedge microcontroller to control the opening and closing of the switching device so that the direct connection between the keyboard microcontroller and the personal computer microcontroller may be disabled when the wedge microcontroller has data to be transmitted to the personal computer microcontroller.

In accordance with a third aspect of the present invention, a method of transmitting data from a wedge microcontroller to a personal computer microcontroller is disclosed, wherein the personal computer microcontroller is directly connected to a keyboard microcontroller by means of a clock line and a data line with a switching device positioned therebetween. The wedge microcontroller includes a first set of connections to the clock and data lines between the switching device and the keyboard microcontroller and a second set of connections to the clock and data lines between the switching device and the personal computer microcontroller. The method includes the steps of: determining whether the wedge microcontroller has data to be transmitted to the personal computer microcontroller; monitoring the data line between the keyboard microcontroller and the personal computer microcontroller, opening the switching device when no activity is detected on the clock line or the data line; and, transmitting data from the wedge microcontroller to the personal computer microcontroller.

BRIEF DESCRIPTION OF THE DRAWING

While the specification concludes with claims particularly pointing out and distinctly claiming the present invention, it is believed the same will be better understood from the following description taken in conjunction with the accompanying drawing in which:

FIG. 1 is a schematic diagram of a prior art system interfacing a peripheral device with a personal computer and its keyboard;

FIG. 2 is a schematic diagram of a system interfacing a peripheral device with a personal computer and its keyboard in accordance with the present invention; and

FIGS. 3A and 3B are a flow diagram depicting the process of transmitting data from the peripheral device shown in FIG. 2 to the personal computer.

DETAILED DESCRIPTION OF THE INVENTION

Referring now to the drawing in detail, wherein identical numerals indicate the same elements throughout the figures, FIG. 1 depicts an electronic cash register system having peripherals interfaced therewith as disclosed in U.S. Pat. No. 5,179,375. As seen therein, the keyboard 12 of electronic cash register 10 transmits signals indirectly to terminal 11 via a decoder (wedge) 14. An external device 15 also is connected to decoder (wedge) 14 so that decoder (wedge) 14 must determine which signal to send to terminal 11 of electronic cash register 10.

Contrary to the system depicted in FIG. 1, the present invention involves a system, designated generally by the numeral **20**, in which a personal computer **22** (or more specifically the microcontroller thereof) is directly connected to a keyboard **24** (or more specifically the microcontroller thereof). In particular, personal computer microcontroller **22** is connected directly to keyboard microcontroller **24** by means of a standard interface in which only a clock line **26** and a data line **28** are shown.

A switching device **30** is provided between personal computer microcontroller 22 and keyboard microcontroller **24** on both clock and data lines **26** and **28**, respectively. Thus, while switching device **30** will normally be in the closed position, whereby data may be transmitted directly from keyboard microcontroller **24** to personal computer microcontroller **22**, this connection may be electrically disabled by opening switching device **30**. It should be noted that a preferred embodiment for switching device **30** is a double-pole, single throw (DPST) silicon switch, consisting of two pairs of matching transistors.

In order that at least one peripheral device **32** may transmit data to personal computer microcontroller **22**, a wedge microcontroller **34** is provided in system **20**. While the present invention is not limited by the use of a particular wedge microcontroller, an exemplary wedge microcontroller is the "Wedge Dynakey Control Board" sold by NCR Corp. in Dayton, Ohio. It will be seen that wedge microcontroller **34** has a first set of connections **36** and **38** to clock and data lines **26** and **28**, respectively, which are positioned between switching device **30** and keyboard microcontroller **24**. Likewise, wedge microcontroller **34** has a second set of connections **40** and **42** to clock and data lines **26** and **28**, respectively, which are positioned between switching device **30** and personal computer microcontroller **22**. A switch control connection **44** is also provided between wedge microcontroller **34** and switching device **30** so that wedge microcontroller **34** is able to control the opening and closing of switching device **30** and therefore whether personal computer microcontroller **22** and keyboard microcontroller **24** remain connected. In this way, wedge microcontroller **34** periodically enables data to be transmitted from peripheral device **32** to personal computer microcontroller **22**.

It is preferred that switching device **30** be biased closed to maintain the direct connection between keyboard microcontroller **24** and personal computer microcontroller **22**. Accordingly, system **20** is designed so that switching device **30** will be closed upon powering up. In order for the direct connection between keyboard microcontroller **24** and personal computer microcontroller **22** to be disabled, affirmative action must be taken by wedge microcontroller **34** to open switching device **30**. This action involves the transmission of a signal from wedge microcontroller **34** to switching device **30** via switch signal connection **44**.

Before wedge mechanism **34** transmits a signal to open switching device **30**, however, wedge mechanism **34** must determine first whether peripheral device **32** has data to be sent to personal computer microcontroller **22**. If so, wedge microcontroller must also determine whether there is any activity on data line **28** between keyboard microcontroller **24** and personal computer microcontroller, which is accomplished by monitoring data line **28** through connection **38**. This is due to the preference of having personal computer microcontroller **22** receive data from only one source at a given time. Provided there is no activity on data line **28** (i.e., it is at an idle state), then wedge microcontroller **34** signals switching device **30** to open and disable the direct electrical connection between personal computer microcontroller **22** and keyboard microcontroller **24**.

Once switching device **30** is opened, wedge microcontroller **34** is connected to personal computer microcontroller

22 in a manner which looks (to personal computer microcontroller 22) like keyboard microcontroller 24. Likewise, wedge microcontroller 34 is connected to keyboard microcontroller 24 in a manner which looks (to keyboard microcontroller 24) like personal computer microcontroller 22. At this point, wedge microcontroller 34 is free to transmit data to personal computer microcontroller 22 from peripheral device 32. In fact, it is preferred that the opening of switching device 30 and transmission of data from wedge microcontroller 34 to personal computer microcontroller 22 occur substantially simultaneously in order to maintain the aforementioned appearances of wedge microcontroller 34 to keyboard microcontroller 24 and personal computer microcontroller 22.

In order to prevent keyboard microcontroller 24 from attempting to transmit data to personal computer microcontroller 22 while data is being transmitted from wedge microcontroller 34 thereto, wedge microcontroller 34 sends a signal to clock line 26 via connection 36 which puts clock line 26 in a low condition. This, in turn, places keyboard microcontroller 24 in a hold state. During this hold state, keyboard microcontroller 24 preferably stores data inputted therein in a buffer for later transmission to personal computer microcontroller 22. It is also preferred that the opening of switching device 30 (and therefore the electrical disabling of the connection between personal computer microcontroller 22 and keyboard microcontroller 24) and the preventing of data from being transmitted by keyboard microcontroller 24 (by placing it in a hold state) occur substantially simultaneously.

After wedge microcontroller 34 has completed the transmission of data from peripheral device 32 to personal computer microcontroller 22, wedge microcontroller 34 sends a signal to switching device 30 via switch control connection 44 which closes switching device 30 and reconnects personal computer microcontroller 22 and keyboard microcontroller 24. Upon closing of switching device 30, keyboard microcontroller 24 is released from its hold state by wedge microcontroller 34 so that it is able to transmit data (including that stored in a buffer during data transmission by wedge microcontroller 34) to personal computer microcontroller 22.

It will therefore be understood from the above description that personal computer microcontroller 22 will remain directly connected to keyboard microcontroller 24 unless switching device 30 is opened. Also, when switching device 30 is in the closed position, personal computer microcontroller 22 and keyboard microcontroller 24 will function without regard to wedge microcontroller 34. This type of connection between personal computer microcontroller 22, keyboard microcontroller 24, and wedge microcontroller 34, in which switching device 30 is able to dynamically disconnect and reconnect personal computer microcontroller 22 and keyboard microcontroller 24, minimizes the amount of interference between personal computer microcontroller 22 and keyboard microcontroller 24 (and therefore timing problems therebetween). It also limits the amount of exposure wedge microcontroller 34 has with respect to modifications in keyboard communications since relatively little keyboard functionality is required within wedge microcontroller 34.

Further, as shown in the flow diagram of FIG. 3, system 20 operates in accordance with a specified process before wedge microcontroller 34 transmits data of peripheral device 32 to personal computer microcontroller 22. This process includes the steps of determining whether there is data to be transmitted to personal computer microcontroller 22 from wedge microcontroller 34, monitoring data line 28 between keyboard microcontroller 24 and personal computer microcontroller 22, opening switching device 30 when no activity is detected on data line 28, and transmitting data from wedge microcontroller 34 to personal computer microcontroller 22.

Having shown and described the preferred embodiment of the present invention, further adaptations of the above-described system for interfacing at least one peripheral device with a personal computer, as well as the wedge microcontroller utilized therefor, can be accomplished by appropriate modifications by one of ordinary skill in the art without departing from the scope of the invention which includes the claimed methods.

What is claimed is:

1. A system for interfacing at least one peripheral device with a personal computer microcontroller, comprising:
  - (a) a personal computer microcontroller;
  - (b) a keyboard microcontroller directly connected to said personal computer microcontroller by means of a clock line and a data line;
  - (c) a switching device positioned between said personal computer microcontroller and said keyboard microcontroller; and
  - (d) a wedge microcontroller to which said peripheral device is connected, comprising:
    - (1) a first set of connections to said clock and data lines positioned between said switching device and said keyboard microcontroller;
    - (2) a second set of connections to said clock and data lines positioned between said switching device and said personal computer microcontroller; and
    - (3) a switching device control connection to said switching device for opening and closing said switching device;
  - wherein the direct connection between said personal computer microcontroller and said keyboard microcontroller is disconnected by opening said switching device so that said wedge microcontroller is able to transmit data to said personal computer microcontroller;
  - wherein said keyboard microcontroller is actively prevented from transmitting data to said personal computer microcontroller during transmission of data from said wedge microcontroller to said personal computer microcontroller, by said wedge microcontroller sending a signal to said keyboard microcontroller via said clock line that places said keyboard microcontroller in a hold state, for as long as the wedge microcontroller is transmitting data to said personal computer microcontroller.

2. The system of claim 1, wherein data inputted to said keyboard microcontroller is stored during said hold state.

3. The system of claim 1, wherein said wedge microcontroller sends a signal via said switch control connection to open said switching device when said wedge microcontroller has data to be transmitted to said personal computer microcontroller.

4. The system of claim 1, wherein said wedge microcontroller monitors said clock and data lines to determine whether data is being transmitted from said keyboard microcontroller to said personal computer microcontroller without affecting the direct connection between said keyboard and personal computer microcontrollers.

5. The system of claim 1, wherein opening of said switching device and preventing said keyboard microcontroller from transmitting data to said personal computer microcontroller occur substantially simultaneously.

6. The system of claim 1, wherein said personal computer microcontroller and said keyboard microcontroller function without regard to said wedge microcontroller when said switching device is closed.

7. The system of claim 1, wherein said personal computer microcontroller and said keyboard microcontroller remain directly connected unless said switching device is opened.

8. The system of claim 1, wherein said switching device is utilized to dynamically disconnect and reconnect said personal computer microcontroller and said keyboard microcontroller.

9. The system of claim 4, wherein said wedge microcontroller sends a signal via said switch control connection to open said switching device when said clock and data lines between said keyboard microcontroller and said personal computer microcontroller are determined to be idle.

10. The system of claim 3, wherein said wedge microcontroller sends a signal via said switch control connection to close said switching device upon completion of data transmission from said wedge microcontroller to said personal computer microcontroller so that said keyboard microcontroller and said personal computer microcontroller are reconnected.

11. The system of claim 3, wherein opening of said switching device and transmission of data from said wedge microcontroller to said personal computer microcontroller occur substantially simultaneously.

12. A method of periodically connecting a wedge microcontroller to a personal computer microcontroller for data transmission, wherein said personal computer microcontroller is directly connected to a keyboard microcontroller by means of a clock line and a data line, said method comprising the following steps:

(a) providing a switching device in the clock and data lines connecting said personal computer microcontroller and said keyboard microcontroller;

(b) providing said wedge microcontroller with a first set of connections to said clock and data lines at a location between said switching device and said keyboard microcontroller;

(c) providing said wedge microcontroller with a second set of connections to said clock and data lines at a location between said switching device and said personal computer microcontroller;

(d) permitting said wedge microcontroller to control the opening and closing of said switching device so that said direct connection between said keyboard microcontroller and said personal computer microcontroller may be disabled when said wedge microcontroller has data to be transmitted to said personal computer microcontroller; and

(e) actively preventing said keyboard microcontroller from transmitting data to said personal computer microcontroller during transmission of data from said wedge microcontroller to said personal computer microcontroller, by said wedge microcontroller sending a signal to said keyboard microcontroller via said clock line that places said keyboard microcontroller in a hold state, for as long as the wedge microcontroller is transmitting data to said personal computer microcontroller.

13. A method of transmitting data from a wedge microcontroller to a personal computer microcontroller, wherein said personal computer microcontroller is directly connected to a keyboard microcontroller by means of a clock line and a data line with a switching device positioned therebetween, said wedge microcontroller including a first set of connections to said clock and data lines between said switching device and said keyboard microcontroller and a second set of connections to said clock and data lines between said switching device and said personal computer microcontroller, said method comprising the steps of:

(a) determining whether said wedge microcontroller has data to be transmitted to said personal computer microcontroller;

(b) monitoring said data line between said keyboard microcontroller and said personal computer microcontroller;

(c) opening said switching device when no activity is detected on said data line;

(d) transmitting data from said wedge microcontroller to said wedge microcontroller to said personal computer microcontroller; and

(e) actively preventing said keyboard microcontroller from transmitting data to said personal computer microcontroller during transmission of data from said wedge microcontroller to said personal computer microcontroller, by said wedge microcontroller sending a signal to said keyboard microcontroller via said clock line that places said keyboard microcontroller in a hold state, for as long as the wedge microcontroller is transmitting data to said personal computer microcontroller.

14. The method of claim 13, wherein opening of said switching device and transmission of data from said wedge microcontroller to said personal computer microcontroller occur substantially simultaneously.

15. The method of claim 13, further comprising the step of closing said switching device when said wedge microcontroller has completed transmission of data to said personal computer microcontroller.

16. The method of claim 13, further comprising the step of storing data inputted to said keyboard microcontroller when said switching device is open.

17. The method of claim 16, further comprising the step of transmitting said stored data from said keyboard microcontroller directly to said personal computer microcontroller after said switching device is reclosed.

* * * * *

US006504626B1

(12) **United States Patent**
Shih

(10) **Patent No.: US 6,504,626 B1**
(45) **Date of Patent: Jan. 7, 2003**

(54) **SCANNER WITH AN EXTERNAL KEYBOARD FOR CONTROLLING OPERATIONS OF THE SCANNER**

(75) Inventor: **Po-Sheng Shih**, Hsin-Chu (TW)

(73) Assignee: **eMemory Technology Inc.**, Hsin-Chu (TW)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/378,702**

(22) Filed: **Aug. 23, 1999**

(30) **Foreign Application Priority Data**

Jul. 26, 1999 (TW) .... 88112606 A

(51) **Int. Cl.**^{7} **H04N 1/32**

(52) **U.S. Cl.** **358/442**; 358/497; 358/474; 358/468; 361/686; 361/679; 361/680; 361/681; 710/73; 710/5

(58) **Field of Search** 358/442, 474, 358/468, 494, 400, 421, 497; 361/686, 679, 680, 681; 710/8, 72, 73, 5; 235/435; 382/312, 325, 313

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,571,053 A * 2/1986 Kasama et al.
4,893,196 A * 1/1990 Koshiyouji et al.
5,091,747 A * 2/1992 Tsai
5,550,938 A * 8/1996 Hayakawa et al.
5,555,105 A * 9/1996 Shahir et al.
5,822,080 A * 10/1998 Chavez
5,854,945 A * 12/1998 Criscito et al.
5,920,401 A * 7/1999 Street et al.
5,928,337 A * 7/1999 Wieringa
6,078,407 A * 6/2000 Ma
6,101,004 A * 8/2000 Sung
6,233,065 B1 * 5/2001 Lee
6,312,175 B1 * 11/2001 Lum

FOREIGN PATENT DOCUMENTS

CN 1284688 * 2/2002

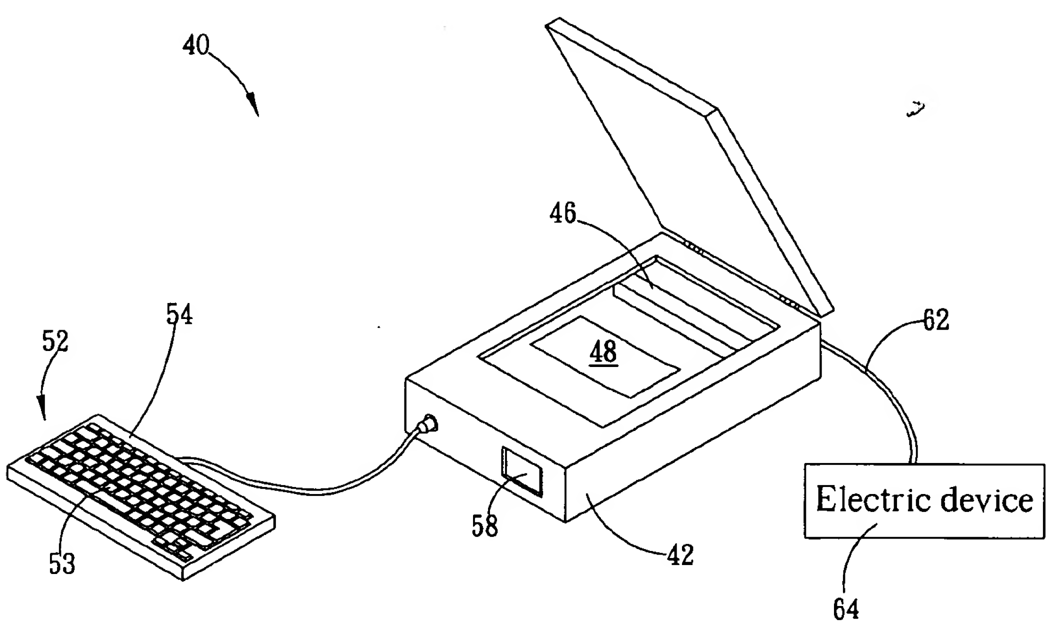
* cited by examiner

Primary Examiner—Edward Coles
Assistant Examiner—Tia A. Carter
(74) *Attorney, Agent, or Firm*—Winston Hsu

(57) **ABSTRACT**

The invention relates to a scanner with an external keyboard. The scanner comprises a scanner housing, a control circuit installed in the scanner housing for controlling operations of the scanner, a scanning module installed in the scanner housing and connected to the control circuit for scanning a document and generating corresponding document image signals, a keyboard electrically connected to the control circuit for inputting various key signals, and a display panel installed on the surface of the scanner housing and connected to the control circuit for displaying signals transmitted from the control circuit. The control circuit controls the operations of the scanner according to the key signals inputted by a user through the keyboard, and displays the key signals or instruction messages on the display panel.

15 Claims, 4 Drawing Sheets



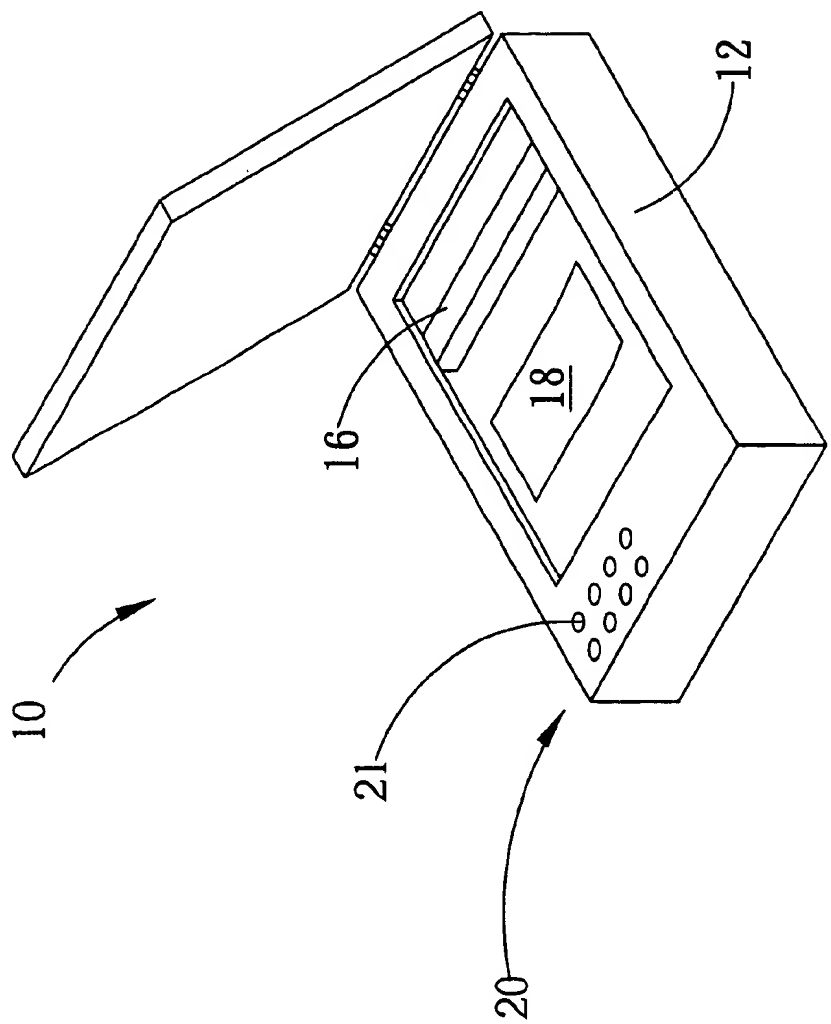


Fig. 1 Prior art

10/29/2003, EAST Version: 1.4.1

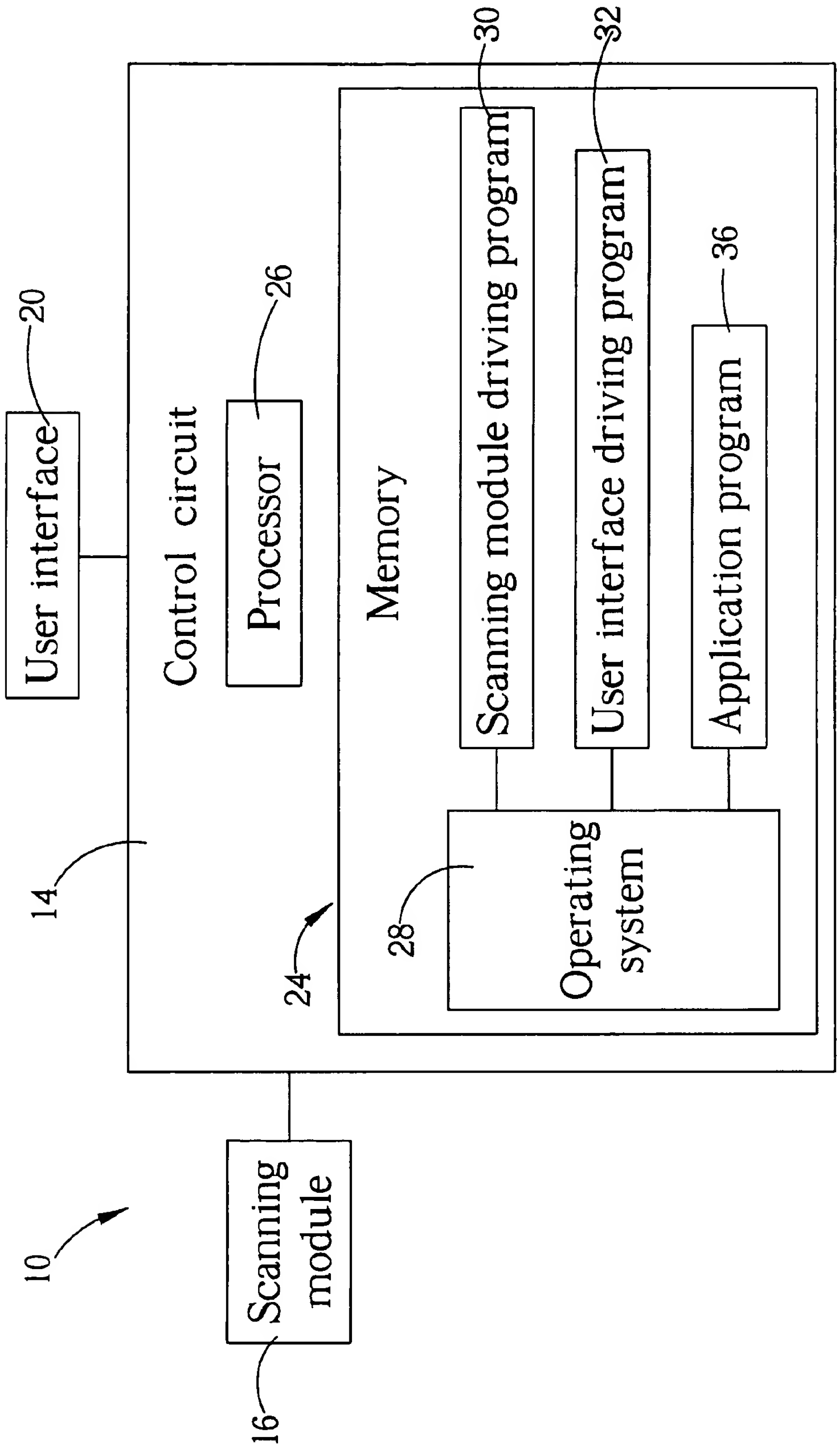


Fig. 2 Prior art

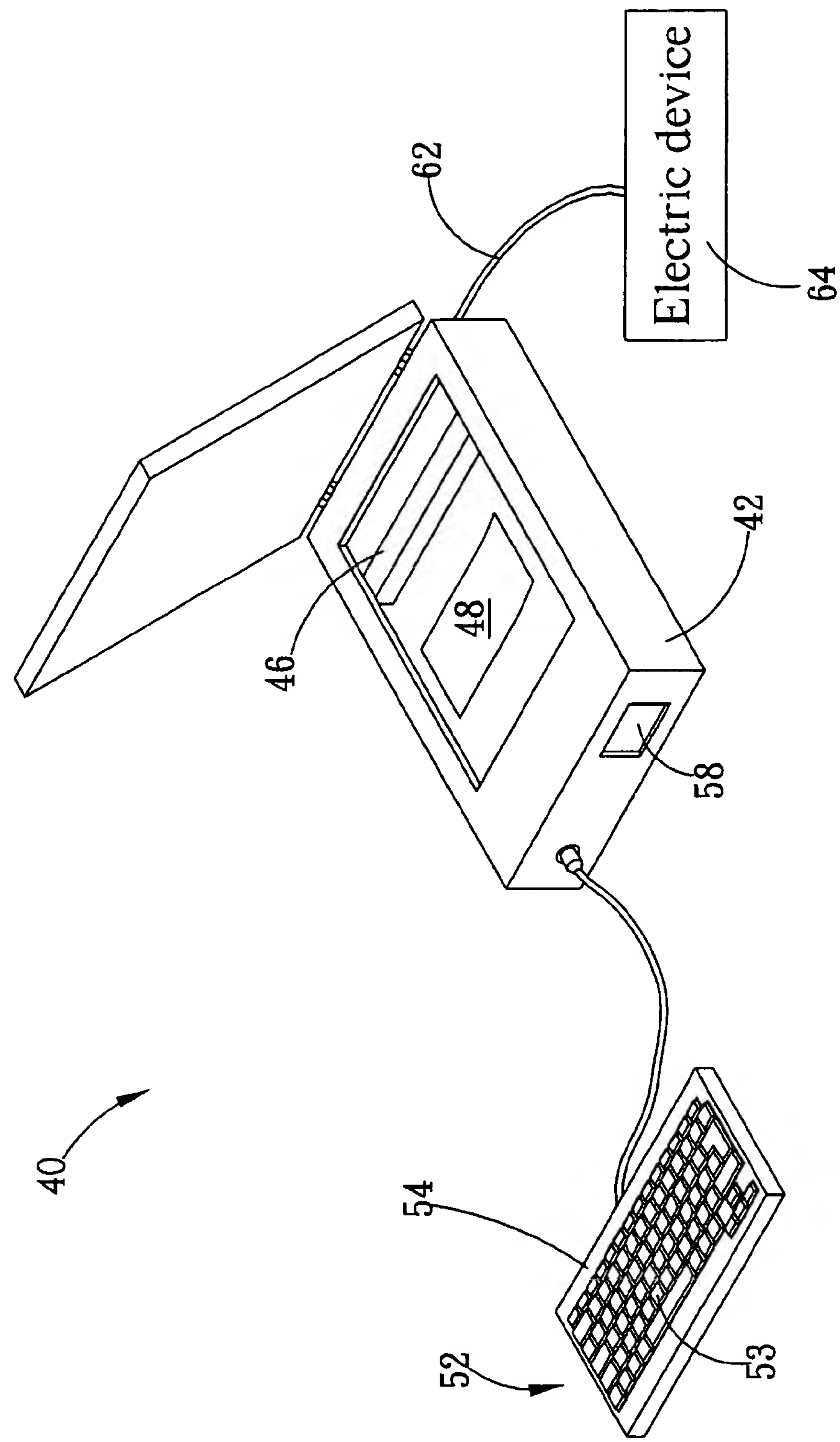


Fig. 3

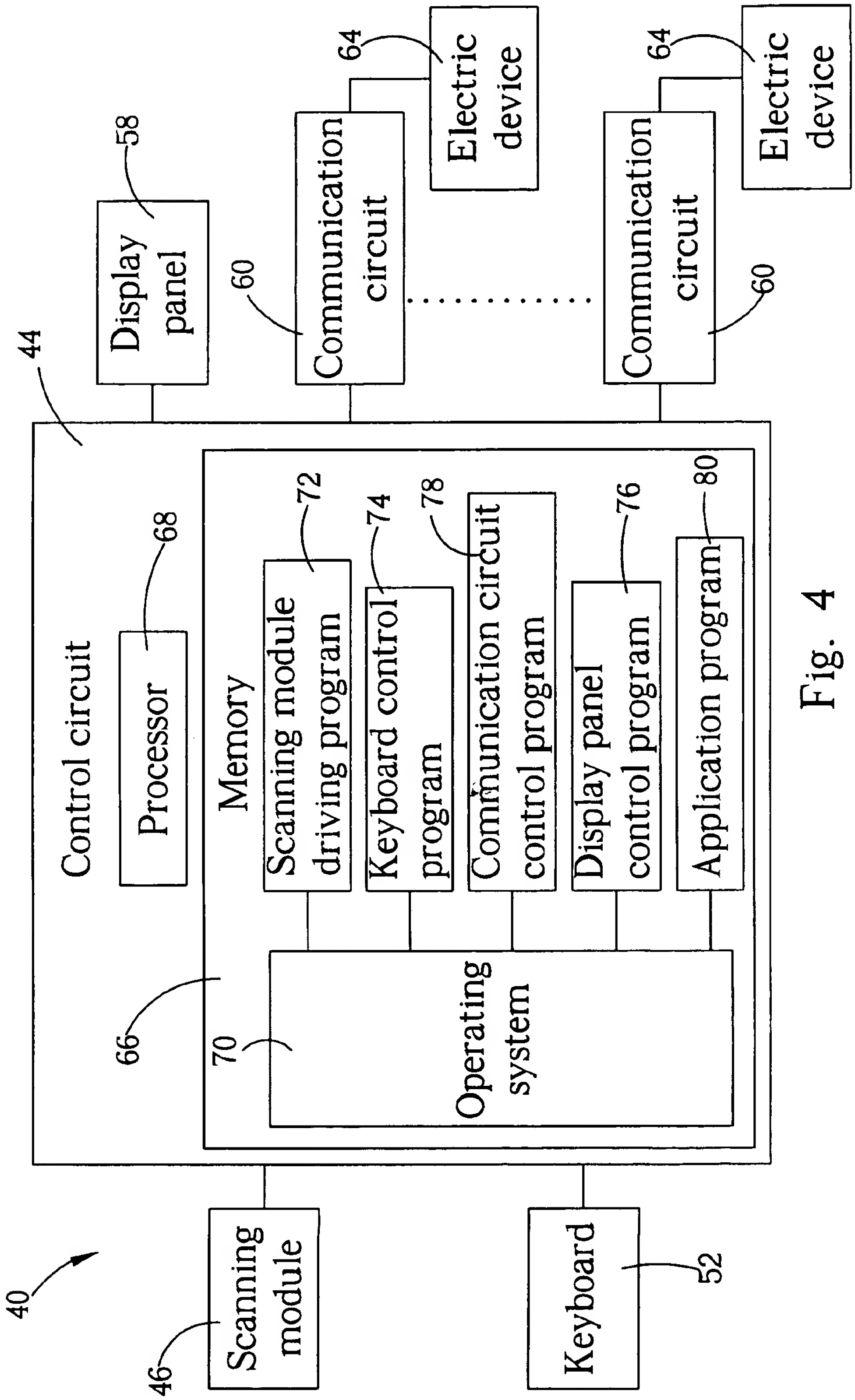

40
Scanning module
46
Keyboard
52
Control circuit
44
Processor
68
Memory
66
Operating system
70
Scanning module driving program
72
Keyboard control program
74
Communication circuit control program
78
Display panel control program
76
Application program
80
Display panel
58
Communication circuit
60
Electric device
64
Communication circuit
60
Electric device
64

Fig. 4

SCANNER WITH AN EXTERNAL KEYBOARD FOR CONTROLLING OPERATIONS OF THE SCANNER

BACKGROUND OF THE INVENTION

1. Field of the Invention

The invention relates to a scanner, and more particularly, to a scanner with an external keyboard for controlling operations of the scanner.

2. Description of the Prior Art

Please refer to FIG. **1** and FIG. **2**. FIG. **1** is a perspective diagram of a prior art scanner **10**. FIG. **2** is a functional block diagram of the scanner **10** in FIG. **1**. The scanner **10** comprises a scanner housing **12**, a control circuit **14** installed in the scanner housing **12** for controlling operations of the scanner **10**, a scanning module **16** installed in the scanner housing **12** and connected to the control circuit **14** for scanning a document and generating corresponding document image signals, and a user interface **20** installed on the scanner housing **12** and connected to the control circuit **14**. The user interface **20** comprises a plurality of keys **21** for inputting key signals.

The control circuit 14 comprises a memory **24** for storing programs and data, and a processor **26** for executing the programs in the memory **24**. The memory **24** comprises an operating system **28** for controlling the operations of the processor **26**, a scanning module driving program **30** executed under the operating system **28** for controlling the operations of the scanning module **16** and processing the document image signals generated by the scanning module **16**, a user interface driving program **32** executed under the operating system **28** for controlling the operations of the user interface **20**, and an application program **36** executed under the operating system **28** for setting the operation mode of the scanning module driving program **30** and the processing method of the image signals according to the key signals inputted by the user and transmitted by using the user interface driving program **32**.

The control circuit **14** controls the operations of the scanner **10** according to the key signals inputted by the user and transmitted from the user interface **20**. Because the space on the scanner housing **12** is very limited, the scanner housing **12** is unable to accommodate sufficient number of regular sized keys **21**. Unless the keys **21** are made very small, the small number of keys **21** may limit the performance of the scanner **10**. Moreover, it is very costly to install a large number of small keys **21** on the scanner **10**. And it will make the scanner **10** look cumbersome if a great number of large keys **21** are installed on the scanner **10**.

SUMMARY OF THE INVENTION

It is therefore a primary objective of the present invention to provide a scanner with an external keyboard to solve the mentioned problem.

Briefly, in a preferred embodiment, the present invention provides a scanner comprising:

- a scanner housing;
- a control circuit installed in the scanner housing for controlling operations of the scanner;
- a scanning module installed in the scanner housing and connected to the control circuit for scanning a document and generating corresponding document image signals;
- a keyboard connected to the control circuit for inputting key signals; and
- a display panel installed on the surface of the scanner housing and connected to the control circuit for displaying signals transmitted from the control circuit;

wherein the control circuit controls the operations of the scanner according to the key signals inputted by a user through the keyboard, and displays the key signals or instruction messages on the display panel.

It is an advantage of the present invention that the scanner has an external keyboard. Therefore various key signals can be easily inputted through regular sized keys on the keyboard to control operations of the scanner.

These and other objects and the advantages of the present invention will no doubt become obvious to those of ordinary skill in the art after having read the following detailed description of the preferred embodiment that is illustrated in the various figures and drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a perspective diagram of a prior art scanner.

FIG. **2** is a functional block diagram of the scanner in FIG. **1**.

FIG. **3** is a perspective diagram of a scanner according to the present invention.

FIG. **4** is a functional block diagram of the scanner in FIG. **3**.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Please refer to FIG. **3** and FIG. **4**. FIG. **3** is a perspective diagram of a scanner **40** according to the present invention. FIG. **4** is a functional block diagram of the scanner **40** in FIG. **3**. The scanner **40** comprises a scanner housing **42**, a control circuit **44** installed in the scanner housing **42** for controlling operations of the scanner **40**, a scanning module **46** installed in the scanner housing **42** and connected to the control circuit **44** for scanning a document **48** and generating corresponding document image signals, a keyboard **52** electrically connected to the control circuit **44** for inputting the key signals, a display panel **58** installed on the surface of the scanner housing **42** and connected to the control circuit **44** for displaying the signals transmitted from the control circuit **44**, and at least one communication circuit **60** for transmitting electric signals to an electric device **64**. The control circuit **44** can be connected to a plurality of communication circuits **60** at the same time so that data can be transmitted to the electric devices **64** connected to each of the communication circuits **60** concurrently. The keyboard **52** comprises a keyboard housing **54**, a plurality of number keys, character keys, symbol keys, control keys and function keys **53** positioned on the keyboard housing **54** for generating key signals. Moreover, the keyboard **52** can be connected with the control circuit **44** in a wireless manner.

The control circuit **44** comprises a memory **66** for storing programs and data, and a processor **68** for executing the programs in the memory **66**. The memory **66** comprises an operating system **70** for controlling the operations of the processor **68**, a scanning module driving program **72** executed under the operating system **70** for controlling the operations of the scanning module **46** and processing the document image signals generated by the scanning module **46**, a keyboard control program **74** executed under the operating system **70** for controlling the operations of the keyboard **52**, a display panel control program **76** executed under the operating system **70** for controlling the operations of the display panel **58**, a communication circuit control program **78** executed under the operating system **70** for controlling the operations of the communication circuit **60**, and an application program **80** executed under the operating system **70**.

10/29/2003, EAST Version: 1.4.1

The application program **80** is used for setting the operation mode of the scanning module driving program **72** and the processing method of the image signals according to the key signals inputted by the user and transmitted from the keyboard control program **74**, displaying the key signals or the instruction messages on the display panel **58** by using the display panel control program **76**, and transmitting the image signals to the electric device **64** by using the communication circuit control program **78**.

The electric device **64** can be an internet server or a local area network (LAN) server, and the application program **80** can be an e-mail processing program for transmitting the document image signals in an e-mail format to the internet or LAN server by using the communication circuit control program **78**. The electric device **64** can be a printer, and the application program **80** can be a document image processing program that transmits the document image signals to the printer for printing by using the communication circuit control program **78**. The electric device **64** can be a personal computer, and the application program **80** can be a document processing program that transmits the document image signals to the personal computer by using the communication circuit control program **78**. The electric device **64** can also be a fax machine, and the application program **80** can be a facsimile processing program that transmits the document image signals to the fax machine by using the communication circuit control program **78**.

The communication circuit **60** can be a modem or a signal modulation circuit. The communication circuit **60** also can be a network card, a network interface circuit, a printer interface circuit, or a computer interface circuit (such as SCSI, EPP, IEEE1284.3 or RS232 and so on) for transmitting the document image signals to the electric device **64**.

Compared with the prior art scanner **10**, the scanner **40** has an external keyboard **52**. Therefore, a large number of functions of the scanner **40** can be easily inputted through the keys **53** on the keyboard **52**. For example, a user can use the keyboard **52** to input an e-mail address, which can be displayed on the display panel **58** as well. By using an e-mail processing program, the scanned image is processed as a file and transmitted to the internet or LAN server through the communication circuit **60** without using a computer. The user also can use the keyboard **52** to input a fax number and transmit the scanned image to a fax machine. Furthermore, the user can use the keyboard **52** to input various parameters of a printer, such as number of pages, size of pages, etc. Moreover, the user has an option to operate the scanner **40** with or without using a computer.

Those skilled in the art will readily observe that numerous modifications and alterations of the device may be made while retaining the teachings of the invention. Accordingly, the above disclosure should be construed as limited only by the metes and bounds of the appended claims.

What is claimed is:

1. A scanner comprising:
   - a scanner housing;
   - a control circuit installed in the scanner housing for controlling operations of the scanner;
   - a scanning module installed in the scanner housing and connected to the control circuit for scanning a document and generating corresponding document image signals;
   - a keyboard electrically connected to the control circuit for inputting key signals; and
   - a display panel installed on the surface of the scanner housing and connected to the control circuit for displaying signals transmitted from the control circuit;

   wherein the control circuit controls the operations of the scanner according to the key signals inputted by a user

   through the keyboard, and displays the key signals or instruction messages on the display panel.

2. The scanner of claim **1** wherein the control circuit comprises a memory for storing programs and data, and a processor for executing the programs in the memory, the memory comprises:
   - an operating system for controlling the operations of the processor;
   - a scanning module driving program executed under the operating system for controlling the operations of the scanning module and processing the document image signals generated by the scanning module;
   - a keyboard control program executed under the operating system for controlling the operations of the keyboard;
   - a display panel control program executed under the operating system for controlling the operations of the display panel; and
   - an application program executed under the operating system for setting the operation mode of the scanning module driving program and the processing method of the image signals according to the key signals inputted by the user and transmitted from the keyboard control program, and displaying the key signals or the instruction messages on the display panel by using the display panel control program.

3. The scanner of claim **2** further comprising a communication circuit for transmitting electric signals to an electric device, and a communication circuit control program stored in the memory for controlling the operations of the communication circuit, wherein the application program transmits the document image signals to the electric device by using the communication circuit control program.

4. The scanner of claim **3** wherein the electric device is an internet server.

5. The scanner of claim **4** wherein the application program is an e-mail processing program for transmitting the document image signals in an e-mail format to the internet server by using the communication circuit control program.

6. The scanner of claim **3** wherein the electric device is a local area network (LAN) server.

7. The scanner of claim **6** wherein the application program is an e-mail processing program for transmitting the document image signals in an e-mail format to the LAN server by using the communication circuit control program.

8. The scanner of claim **3** wherein the electric device is a printer, and the application program is a document image processing program that transmits the document image signals to the printer for printing by using the communication circuit control program.

9. The scanner of claim **3** wherein the electric device is a personal computer, and the application program is a document processing program that transmits the document image signals to the personal computer by using the communication circuit control program.

10. The scanner of claim **3** wherein the electric device is a fax machine.

11. The scanner of claim **10** wherein the application program is a facsimile processing program that transmits the document image signals to the fax machine by using the communication circuit control program.

12. The scanner of claim **3** wherein the communication circuit is a modem or a signal modulation circuit.

13. The scanner of claim **3** wherein the communication circuit is a network card or a network interface circuit.

14. The scanner of claim **3** wherein the communication circuit is a printer interface circuit.

15. The scanner of claim **3** wherein the communication circuit is a computer interface circuit.

* * * * *

US006612495B2

(12) **United States Patent**
Reddersen et al.

(10) **Patent No.: US 6,612,495 B2**
(45) **Date of Patent: *Sep. 2, 2003**

(54) **MULTIPLE-INTERFACE SELECTION SYSTEM FOR COMPUTER PERIPHERALS**

(75) Inventors: **Brad R. Reddersen**, Eugene, OR (US); **Phillip W. Shepard**, Eugene, OR (US); **Rockie D. Moch**, Eugene, OR (US); **Jon Paul Charles Williams**, Eugene, OR (US)

(73) Assignee: **PSC Scanning, Inc.**, Eugene, OR (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **09/964,253**

(22) Filed: **Sep. 25, 2001**

(65) **Prior Publication Data**

US 2002/0130181 A1 Sep. 19, 2002

Related U.S. Application Data

(63) Continuation of application No. 09/268,263, filed on Mar. 15, 1999, now Pat. No. 6,293,467, which is a division of application No. 08/955,864, filed on Oct. 21, 1997, now Pat. No. 5,905,249, which is a continuation of application No. 08/706,736, filed on Sep. 9, 1996, now Pat. No. 5,703,347, which is a continuation of application No. 08/305,517, filed on Sep. 13, 1994, now Pat. No. 5,563,402, which is a continuation-in-part of application No. 08/039,606, filed on Mar. 25, 1993, now Pat. No. 5,347,113, which is a continuation-in-part of application No. 08/034,189, filed on Mar. 22, 1993, now Pat. No. 5,330,370, which is a continuation of application No. 07/788,267, filed on Nov. 4, 1991, now abandoned.

(51) **Int. Cl.**^{7} **G06F 17/00**; G06F 19/00; G06K 7/10

(52) **U.S. Cl.** **235/462.15**; 235/462.13

(58) **Field of Search** 235/462.13, 462.15, 235/462.07; 439/59, 65, 68, 502, 620

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,543,450 A	9/1985	Brandt	179/2
4,579,407 A	4/1986	Shimada	339/29
4,621,189 A	11/1986	Kumar et al.	235/472
4,678,288 A	7/1987	Lonsdale et al.	350/432
4,686,506 A	8/1987	Farago	340/347
4,694,182 A	9/1987	Howard	250/566
4,699,447 A	10/1987	Howard	350/6.9
4,820,193 A	4/1989	Noorily	439/488
4,861,972 A	8/1989	Elliott et al.	235/462
4,866,257 A	9/1989	Elliott et al.	235/436
4,868,375 A	9/1989	Blanford	235/462
4,889,497 A	12/1989	Riches	439/76
4,902,244 A	2/1990	Endo et al.	439/489

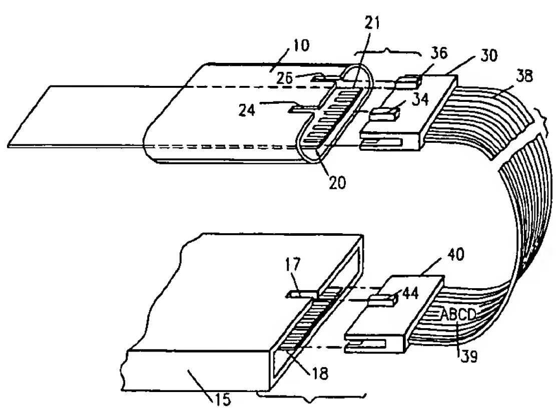
(List continued on next page.)

Primary Examiner—Karl D. Frech
(74) *Attorney, Agent, or Firm*—Stoel Rives LLP

(57) **ABSTRACT**

An interface selection and configuration system for a computer peripheral in which configuration for the peripheral and/or the host interface is at least in part accomplished by the interface connector cable. In a preferred embodiment, the computer peripheral is equipped with one or more hardware interfaces. The interface connector cable has a first end connector for attaching to the computer peripheral. The first end connector of the interface connector cable is typically a multiple pin connector constructed and arranged to be properly physically and electrically connectable only to a specific computer peripheral or class of computer peripherals, the first end connector including at least one electrical connection between two pins for completing a circuit within the computer peripheral thereby enabling the computer peripheral. Where the peripheral is a data reading device such as a laser scanner or RFID reader, alternate or additional configuration may be obtained, with data reading device, from the label on the interconnect cable. The label, which may for example be a bar code or RFID tag, contains information or instructions by which the data reading device (and/or the host) is configured.

16 Claims, 5 Drawing Sheets

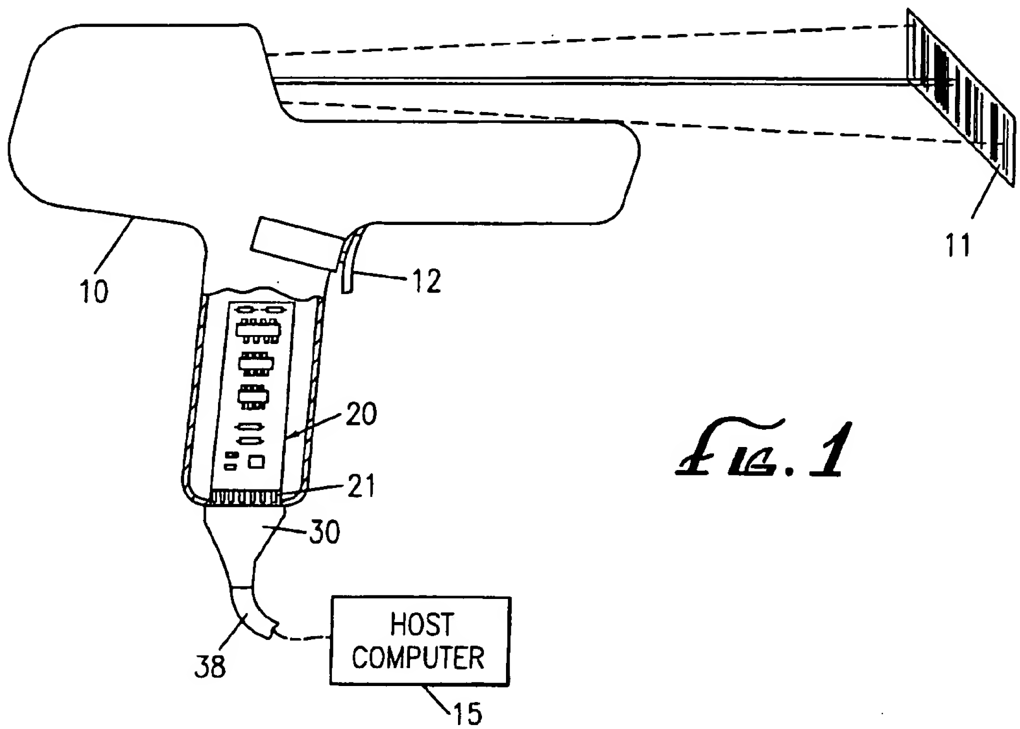


10/29/2003, EAST Version: 1.4.1

U.S. PATENT DOCUMENTS

4,915,639 A	4/1990	Cohn et al.	439/188
4,941,845 A	7/1990	Eppley et al.	439/505
4,954,101 A	9/1990	Nelson	439/502
4,972,470 A	11/1990	Farago	380/3
5,040,993 A	8/1991	Krug et al.	439/75
5,092,793 A	3/1992	Stephan	439/446
5,115,120 A	5/1992	Eastman	235/462
5,181,858 A	1/1993	Matz et al.	439/188
5,189,291 A	2/1993	Siemiatkowski	235/472
5,200,597 A	4/1993	Eastman et al.	235/455
5,214,268 A	5/1993	Doing	235/472
5,218,187 A	6/1993	Koenck et al.	235/675
5,222,164 A	6/1993	Bass, Sr. et al.	385/14
5,239,662 A	8/1993	Danielson et al.	395/800
5,250,792 A	10/1993	Swartz et al.	235/472
5,258,604 A	11/1993	Behrens et al.	235/462
5,305,181 A	4/1994	Schultz	361/680
5,313,053 A	5/1994	Koenck et al.	235/472
5,330,370 A	7/1994	Reddersen et al.	439/502
5,347,113 A	9/1994	Reddersen et al.	235/462
5,563,402 A	10/1996	Reddersen et al.	235/436
5,664,229 A	9/1997	Bhargava et al.	395/885
5,671,374 A	9/1997	Postman et al.	395/309
5,675,139 A	10/1997	Fama	235/472
5,703,347 A	12/1997	Reddersen et al.	235/472
5,745,794 A	4/1998	Poloniewicz et al.	395/882
5,763,865 A	6/1998	Swift et al.	235/472
5,875,415 A	2/1999	Lieb et al.	702/122
5,905,249 A	5/1999	Reddersen et al.	235/462.15
6,036,098 A	3/2000	Goldman et al.	235/486
6,293,467 B1 *	9/2001	Reddersen et al.	235/462.15

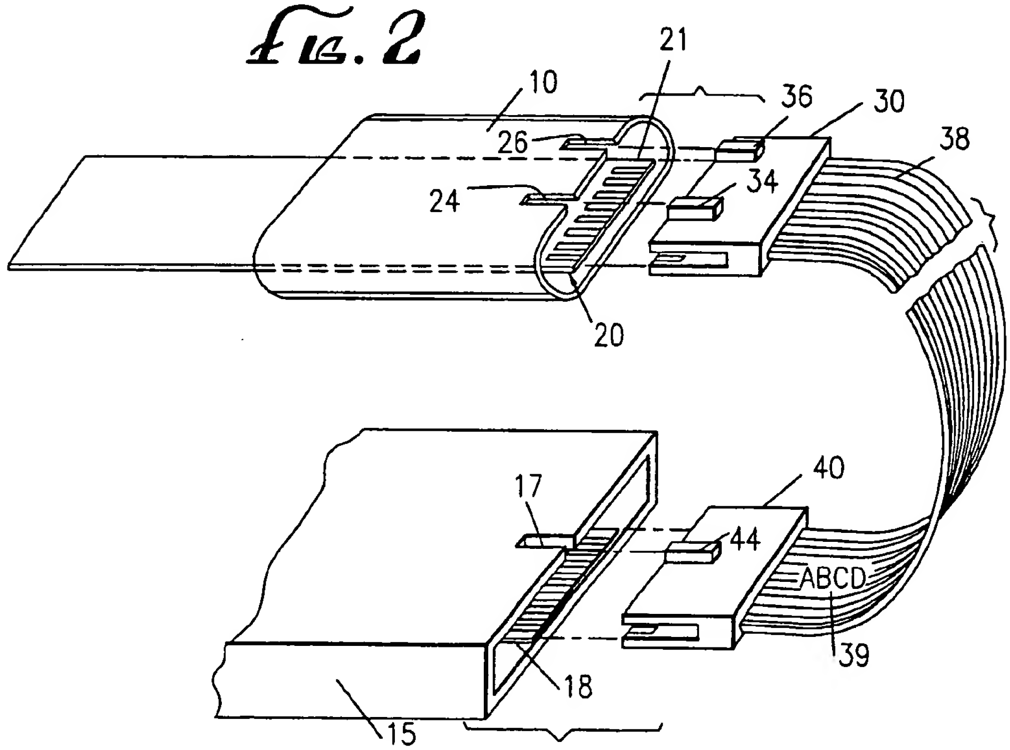
* cited by examiner



11
10
12
20
21
30
38
HOST COMPUTER
15

FIG. 1

FIG. 2



21
10
36
30
38
26
24
34
20
40
17
44
ABCD
39
18
15

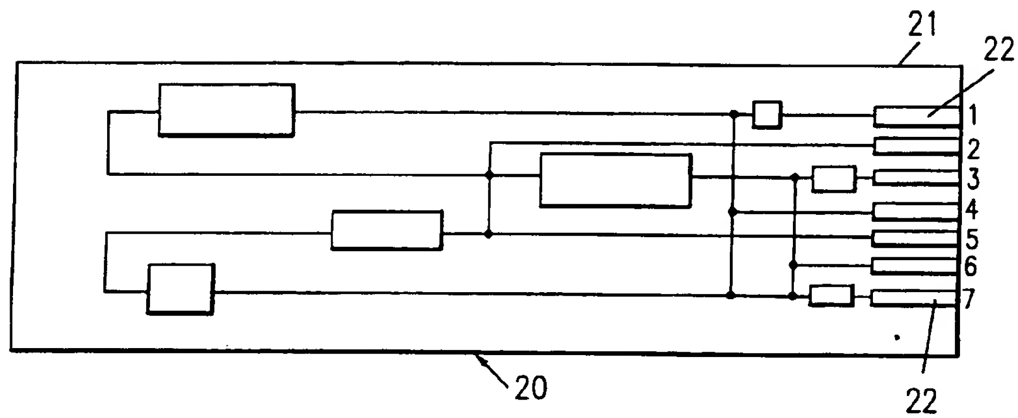


FIG. 3

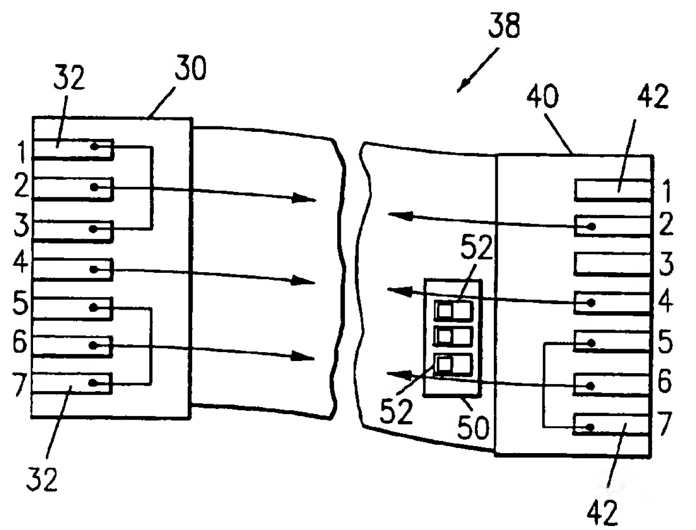


FIG. 4

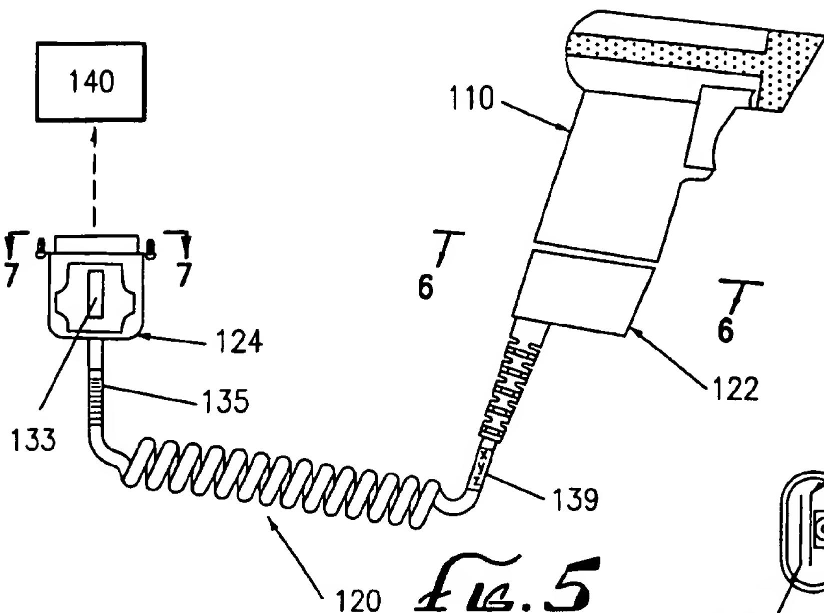


FIG. 5

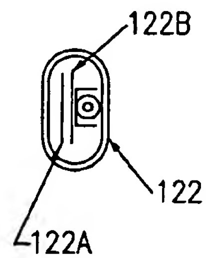


FIG. 6

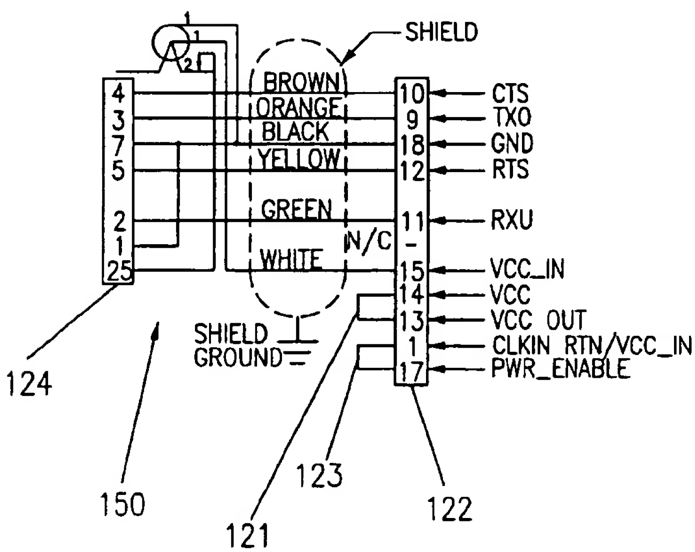


FIG. 8

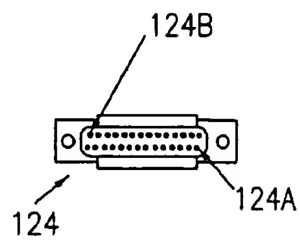


FIG. 7

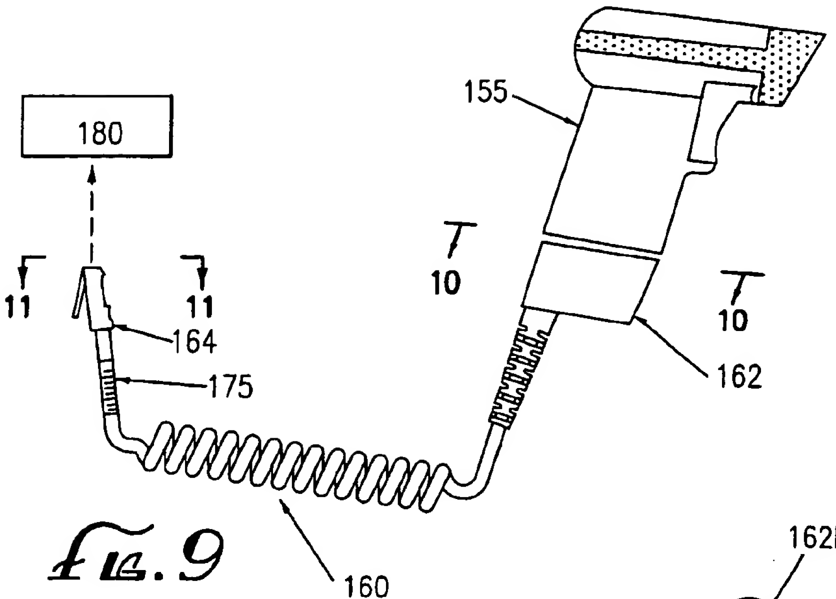


FIG. 9

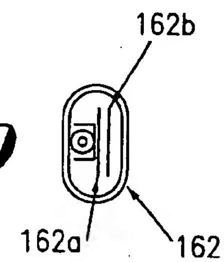
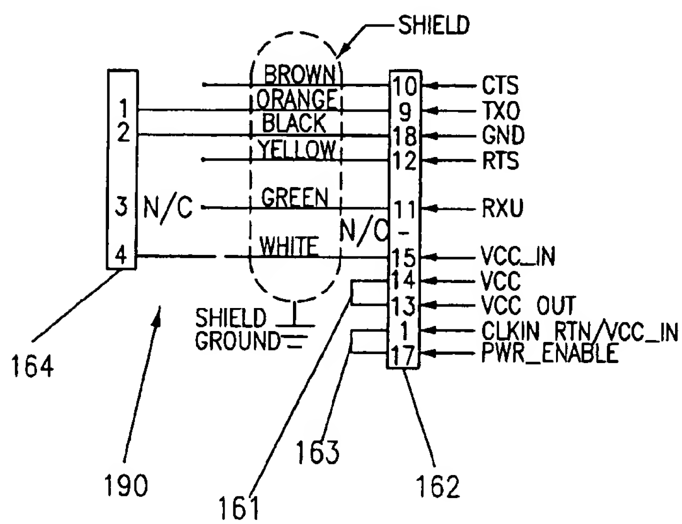


FIG. 10



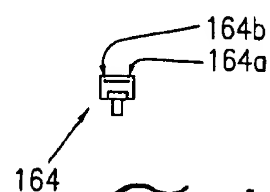


FIG. 11

FIG. 12

10/29/2003, EAST Version: 1.4.1

FIG. 13

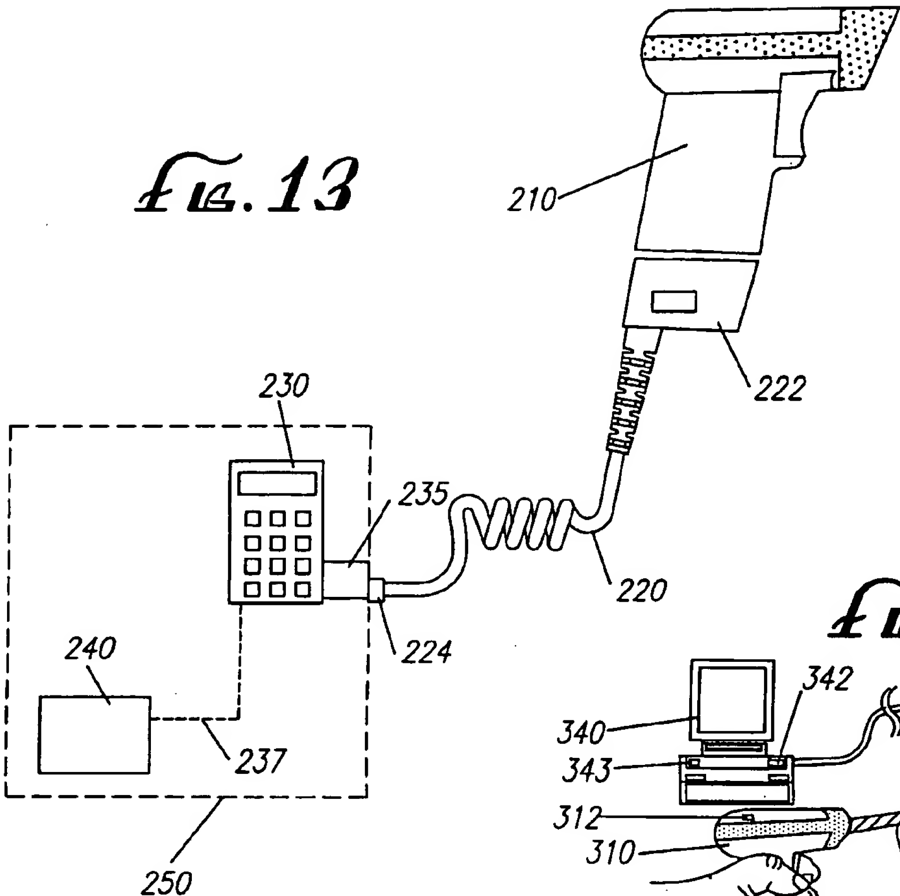
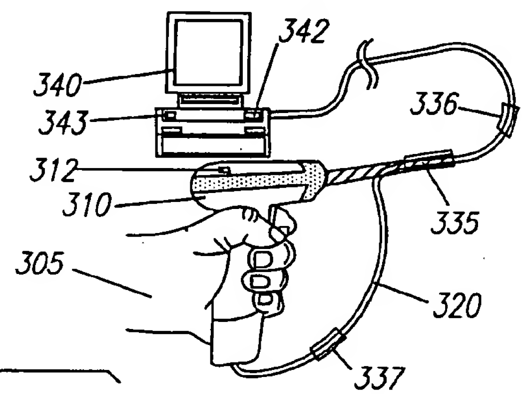


FIG. 14



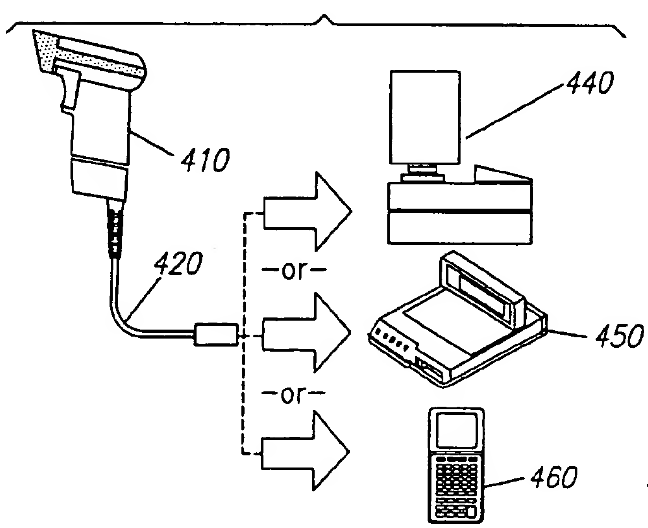


FIG. 15

MULTIPLE-INTERFACE SELECTION SYSTEM FOR COMPUTER PERIPHERALS

RELATED APPLICATIONS

This application is a continuation of Ser. No. 09/268,263 filed Mar. 15, 1999 U.S. Pat. No. 6,293,467, which is a divisional of Ser. No. 08/955,864 filed Oct. 21, 1997 U.S. Pat. No. 5,905,249, which is a continuation of Ser. No. 08/706,736 filed Sep. 9, 1996 U.S. Pat. No. 5,703,347, which is a continuation of Ser. No. 08/305,517 filed Sep. 13, 1994 U.S. Pat. No. 5,563,402, which is a continuation-in-part of application Ser. No. 08/039,606 filed Mar. 25, 1993 U.S. Pat. No. 5,347,113, which is a continuation-in-part of Ser. No. 08/034,189 filed Mar. 22, 1993 U.S. Pat. No. 5,330,370, which is a continuation of application Ser. No. 07/788,267 filed Nov. 4, 1991, abandoned.

BACKGROUND OF THE INVENTION

The field of the present invention relates to interconnection systems for computers and computer peripherals or more specifically, methods and devices for selecting proper interface between a computer peripheral and its host interface such as may include a computer.

Host computers need to be interconnected to a wide variety of peripheral devices including printers, scanners, monitors, and controllers among others. When the host computer is being connected to a certain type of peripheral, for example a handheld laser scanner, the computer typically has a single input/output connector to which the scanner may be connected by an interconnect cable. It is frequently desirable that a particular handheld scanner be usable with a variety of different host computers. Conversely, it is also desirable that the host computer be able to support a variety of different handheld scanners.

Heretofore there have been several systems for achieving proper configuration between the host computer and the peripheral, In a typical system, a particular peripheral is configured to work with a particular host computer or terminal, that is, the peripheral has contained a single dedicated interface. Similarly, the host computer was configured to accept only a particular type of peripheral. Any time the peripheral was moved to a different host computer, it was necessary to replace the interface software and hardware in the peripheral.

The host computer may include a software selection program in which the user inputs information identifying the particular peripheral enabling the system to have proper operation. Such an operation requires the user to correctly input information into the host computer identifying the particular peripheral. Alternately, means are provided for scanning a code on the outside of the peripheral which informs the computer of the type of peripheral. Some peripherals actually include identifying signals which again inform the particular host of the type of peripheral and software provides the desired configuration. Many of these systems still require correct interface hardware.

In another configuration technique, the peripheral includes interface hardware for more than one host computer. When configuring, the printed circuit board of the peripheral requires certain hardware configuration in order to be correctly linked to a host computer. Such hardware configuration may be effectuated by manually actuable external switches or by internal switches or "jumpers" within the printed circuit board (and/or within the host computer) which activate or deactivate certain components. Such an operation typically requires the expertise of an electronics technician or skilled user and is not a desirable field operation to be performed by the typical user. It is desirable to have an inexpensive and easy to use interconnection system which can be effectively used by the average user.

SUMMARY OF THE INVENTION

The present invention relates to an interface selection system for a computer peripheral in which configuration for the peripheral and/or its host is at least in part accomplished through the interface connector cable. In a preferred embodiment, the computer peripheral is equipped with one or more hardware interfaces. The interface connector cable has a first end connector for attaching to the computer peripheral. The first end connector of the interface connector cable is typically a multiple contact connector (such as pin or edge connector) constructed and arranged to be properly physically and electrically connectable only to a specific computer peripheral (or class of computer peripherals), and a specific host interface the cable connector including at least one electrical connection between two contacts for completing a circuit within the computer peripheral thereby enabling the computer peripheral.

In another preferred embodiment where the peripheral is a data reading device such as a laser scanner or an RF identification receiver, alternate or additional configuration may be provided by obtaing, with the data reading device, information from the label on the interconnect cable. The label, which may for example be a bar code, contains information or instructions by which the data reading device (and/or the host) is configured.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a diagrammatic view of a handheld laser scanner attached to a host computer according to the present invention;

FIG. **2** is a diagrammatic view of an interface connector system according to the present invention;

FIG. **3** is a detailed diagrammatic view of a printed circuit board of FIG. **2**;

FIG. **4** is a detailed diagrammatic view of an edge connector as in FIG. **2**;

FIG. **5** is a diagrammatic view of an alternate peripheral configuration system;

FIG. **6** is an end view of an end connector in FIG. **5** taken along line **6—6**;

FIG. **7** is an end view of an end connector in FIG. **5** taken along line **7—7**;

FIG. **8** is a connector schematic illustrating an example cable connection scheme for the cable connector of FIG. **5**;

FIG. **9** is a diagrammatic view of an alternate peripheral configuration system;

FIG. **10** is an end view of an end connector in FIG. **9** taken along line **10—10**;

FIG. **11** is an end view of an end connector in FIG. **9** taken along line **11—11**;

FIG. **12** is a connector schematic illustrating an example cable connection scheme for the cable connector of FIG. **9**;

FIG. **13** illustrates an alternate cable connection embodiment;

FIG. **14** illustrates a cable scanning code operation; and

FIG. **15** illustrates various peripheral/host pairs.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

The preferred embodiments will now be described with reference to the drawings.

In FIG. 1, an example computer peripheral is illustrated as a handheld laser scanner 10 used for scanning a bar code 11. The scanner 10 is operably connected to a host interface (diagrammatically illustrated as a computer 15) via an interconnect cable 38. The interconnect cable 38 includes an end plug or edge connector 30 which connects to a mating connector 21 on the end of a printed circuit board 20 within the scanner 10. The interconnect cable 38 provides a communication link between the host computer 15 and the laser scanner 10 and may also provide power to the scanner 10.

Referring to FIG. 2, the interconnect cable 38 has a first end connector 30 which plugs into the scanner 10 attaching to the edge connector 21 of the circuit board 20, and a second end connector 40 which plugs into the host computer 15 attaching to the edge connector 18. The first end connector 30 includes tabs 34, 36 which slide into and mate with corresponding slots 24, 26 in the body of the handle of the scanner 10. Different types of scanners may be equipped with different positions of the slots 24, 26. Only an end connector 30 having the correct configuration of tabs 34, 36 will be correctly physically connectable to the scanner 10. An interconnect cable 38 without the correct tab configuration cannot physically be plugged into the scanner 10.

Similarly on the host computer side of the interconnect cable 38, the second end connector 40 is equipped with a tab 44 which mates with a corresponding slot 17 at the mating edge connector 18 of the host computer 15. Only an end connector 40 having the correct configuration of the tab 44 will be correctly physically connectable to the host computer 15. An interconnect cable 38 without the correct tab configuration cannot physically be plugged into the host computer 10.

Therefore in order to connect a particular peripheral (such as a laser scanner 10) to a particular host computer, a cable having the correct tab configurations at both end connectors must be selected. The unique physical configurations ensures that the user must select the correct cable for the host computer and peripheral pair. The preferred embodiment may only require tab configuration on the host computer end because it is not anticipated that the various handheld scanners will require the dedicated interconnect cables, but tab configurations on the peripheral end may be desired in other peripheral applications. Further, the tab configurations illustrated are but one example means for ensuring proper interconnect cable selection. The tabs and slots are readily added to conventional end connectors. The design illustrated does not prevent end connectors without tabs from connecting to the peripheral 10 and the host computer 15, but such design may be modified by one skilled in the art (such as by reversing the positions of the slots/tabs) to prevent such connection.

Referring to FIGS. 3 and 4, the printed circuit board (located within the scanner 10) includes an edge connector 21 shown with seven edge contacts 22 (numbered 1 through 7). Though there is a distinction between edge contacts and pins, for the purposes of the present invention, they are interchangeable—a pin connector with its mating plug connector may be interchanged for an edge connector pair or any other suitable electrical contact pair.

The board 20 is designed and constructed to be operable with a number of host computers by way of an externally activated hardware configuration provided by the interconnect cable 38. As viewed in FIG. 4, in the edge connector 30, the edge contacts #5 and #7 are electrically connected, illustrated as being "jumped". Edge contacts #1 and #3 are also jumped while edge contacts #2, #4 and #6 are used for communication. When the interconnect cable 38 is plugged into the printed circuit board 20, edge contacts #1 and #3 of the edge connector 21 are thereby electrically connected, and edge contacts #5 and #7 are also electrically connected. By so electrically connecting these electrical contacts, certain circuits within the circuit board 20 become electrically connected resulting in a desired configuration. The scanner 10 becomes configured to accept a particular host computer 15. The user has configured the scanner merely by plugging in the correct interconnect cable 38.

A similar configuration selection system may alternately or in combination be applied to the host computer connection side. The edge connector 40 has certain of its edge contacts 42 electrically connected, in the illustrated example pins #5 and #7 of the edge connector 40 are jumped. When the edge connector 40 is plugged into the edge connector 18 of host computer 15, edge contacts #5 and #7 within the edge connector 18 become electrically connected, completing a circuit within the host computer 15, thereby configuring the host computer 15 for the particular peripheral. The user may therefore configure the host computer 15 for the particular scanner 10 by merely selecting and plugging in the correct interconnect cable 38.

In practice, the user would be provided with a single peripheral, such as a scanner and several interconnect cables. To connect the scanner, the user would merely select the interconnect cable corresponding to the particular host computer and plug it into the scanner. The interconnect cable would then configure the scanner for the particular host computer. To move the scanner to a different host computer, the user would merely have to switch cables. The interconnect cable 38 may be provided with means for identifying such as identifying markings 39 imprinted directly on the interconnect cable 38 itself, color coding, a label with identification information connected to the cable 38, or the like to assist the user in selecting the correct interconnect cable for the given host.

The illustrated seven edge connector embodiment is a simplified example for a connector design. Electrical contact configurations may of course be more or less than seven contacts (and the two ends need not be the same). In a preferred scanner application, an interconnect cable plug with 30 pins is set forth in Table A as follows:

TABLE A

Pin #	Signal Name	Data Direction(s)
1	TEST_DATA	<-------->
2	READ_DATA_WAND	-------->
3	START_DATA	<--------
4	CLOCKIN	<--------
5	CLOCKOUT	<--------
6	RDATA_RTN	-------->
7	SDATA_RTN	<--------
8	CLOCK_IN_RTN	<--------
9	CLOCK_OUT_RTN	<--------
10	VCC	<--------
11	VCC_OUT	-------->
12	VCC_IN	<--------
13	CLEAR_TO_SEND	
14	TRANSMIT_DATA	
15	RETURN_DATA	
16	RETURN_S	
17	DATA_PLUS	
18	DATA-MINUS	
19	BEEP_IN	
20	PWR_EN	
21	TRIGGER	
22	BAR_CODE_OUT	

TABLE A-continued

Pin #	Signal Name	Data Direction(s)
23	GOOD_READ_IN	
24	START_OF_SCAN	
25	MTR_FAIL	
26	GROUND	
27	CONFIG_1	
28	CONFIG_2	
29	CONFIG_3	
30	CONFIG_4	

The interconnect cable plug may use certain of the pins for communication or power. Some of the pins may be unused and available for other applications, it being desirable that the same pin design be usable for different configurations. The last four pins #27–#30 are dedicated for providing the configuration for the peripheral. The variations of configurations are limitless and may be designed to suit a particular application. The example in Table A is provided in part to show the wide variety of configurations that may be employed. By the cable configuration scheme, the peripheral may be configured setting for example communication baud rate, bit setting (8-bit, 16-bit etc.), parity or some other parameter.

Though particular types of end connectors have been described, the pin connector may be any suitable electrical connector means for providing electrical contact including plugs, pin connectors, sockets, edge connectors and the like. The end connector has multiple contacts, the contacts providing the actual electrical contact surface. The contacts may be pins, edge contacts, plugs, sockets, or any suitable electrical contact element.

The center of the interconnect cable **38** may comprise any suitable transmission medium including a wire (as illustrated), cable, fiber optic cable, radio frequency link, infra red light link, or other transmission medium.

The cable configuration system described may be combined with other configuration systems, some of which have been previously described. For example, the interconnect cable system described herein may be used to automatically select a configuration for a certain class or group of peripherals. The peripheral may additionally include external (or internal) switches identifying the particular peripheral within the group thereby completing the described configuration.

In practice, a computer peripheral such as a laser scanner **10** will be equipped with hardware and firmware so that it may be used with a plurality of different host computers or computer terminals. To provide initial configuration or change configuration when switching host computers, the peripheral is configured merely by selecting the correct interconnect cable **38**. The electrically connected pins in the end connector of the interconnect cable provide the switching necessary within the scanner **10**, activating or deactivating certain circuits, thereby configuring the scanner **10** for the particular host computer.

The interconnect cable **38** may be designed in any suitable manner. In FIGS. **3** and **4** illustrate jump connections between the respective edge contacts **32** of edge connector **30** (or edge contacts **42** of edge connector **40**), other electrical connection mechanisms may be employed. The contacts may be electrically connected by a simple hard wire connection. The interconnect cable **38** itself may include a printed circuit board **50**, preferably in a unitary structure, which may provides the desired electrical connection

between the pins. The interconnect cable **38** (or the printed circuit board **50** thereon) may itself be equipped with dip switches **52** (dual inline package switches) or some other type of switch. An interconnect cable equipped with switches would have certain advantages as only one cable version need be manufactured. The cable type corresponding to a particular peripheral would be selected by setting the switches (by the manufacturer or by the skilled user) and the cable could then stamped with an identifying code **39**.

As described above, the handheld data reader or other computer peripheral is generally connected by a connector cable to a given host or interface. The host or interface is typically a host computer such as a central processing unit (CPU) or other intermediate device which in turn communicates with the CPU. The host may be a communication module, such as an RF transmitter which is provides a radio frequency communication link to the host computer. In such an application, the cable is nonetheless connected to the host computer or CPU albeit through the communication module. If the peripheral is a printer for example, the host may be a network interface, into which the connector cable is plugged, which is in turn connected to the host computer.

There are myriad of potential hosts for a given peripheral. By way of another example, FIG. **13** illustrates a system in which a handheld bar code scanner **210** is connected to a key entry terminal **230**. The interconnect cable **220** has a first end connector **222** plugged into the handheld bar code scanner **210**. The second end connector **224** is actually plugged into a translator module **235** (sometimes called a "wedge") which converts the signal transmitted from the scanner **210** into a signal of the same form as that produced by the key entry terminal **230**. The key entry terminal **230** is in turn connected by a suitable communication link **237** to the central processing unit **240**. In this arrangement, the host is literally the wedge **240** but may be also be considered to comprise a host assembly contained within the dashed-lined box and designated by numeral **250**.

FIGS. **5**–**8** illustrate a preferred alternate embodiment in which the peripheral is a handheld data reading device **110** such as a bar code laser scanner, a CCD reader or other device. The scanner **110** is attached to a host, diagrammatically illustrated as a data terminal **140**, by a connector cable **120** with a first end connector **122** plugged into the scanner **110** and the second end connector **124** plugged into the data terminal **140**. The connector cable **120** is provided with a label **135** which contains encoded data which may be read by the data reader **110**. The label **135** may, for example, be a separate tag on which the bar code is imprinted or the bar code may be imprinted directly on the cable **120** itself. As described below, the label may also comprise an RFID tag containing the pertinent programming data.

As best shown in FIG. **6**, the first end connector **122** is illustrated as an edge connector having a plurality of edge contacts, including contacts **122*a*, 122*b*,** for connection into a corresponding connector in the handle of the scanner **110**. FIGS. **5** and **7** illustrate the second end connector **124** as a pin connector having a plurality of pins (including pins **124*a*, 124*b***) for connection into a corresponding connector in the data terminal **140**.

The cable **120** may also include an identification label **139** which has identification information to assist the user in selecting the correct cable for the particular scanner and host pair. Other or alternate selection means such as color coding may be provided to assist the user in selecting the correct interconnect cable for the particular application.

FIG. **8** schematically illustrates an example cable connection scheme **150** for the cable connector **120** of FIG. **5**. Color

coded cable wires (Brown, Orange, Black, Yellow, Green, White) provide desired electrical communication path between edge contacts (nos. **10**, **9**, **18** etc.) in the first end connector **122** and respective pin contacts (nos. **4**, **3**, **7** etc) in the second end connector **124**. The cable connector **120** also includes peripheral configuration selection by electrical connector **121** (connecting edge contacts **14** and **13**) and by electrical connector **123** (connecting edge contacts **1** and **17**).

FIGS. **9**–**12** illustrate another preferred alternate embodiment in which the peripheral is a handheld data reading device **155** such as a bar code laser scanner, a CCD reader or other device. The scanner **155** is attached to a host (diagrammatically illustrated as a data terminal **180**) by a connector cable **160** with a first end connector **162** plugged into the scanner **155** and the second end connector **164** plugged into the data terminal **180**. The connector cable **160** is provided with a label **175** which contains encoded data which may be read by the data reader **155**. The label **175** may, for example, be a separate tag on which a bar code is imprinted or the bar code label may be imprinted directly on the cable **160** itself. As described in detail below, if required, the user may at least partly configure the scanner **155** merely by scanning the bar code label **175**. An internal operation routine within the scanner **155** then configures the scanner itself on the basis of the configuration information provided by the bar code.

As best shown in FIG. **10**, the first end connector **162** is illustrated as an edge connector having a plurality of edge contacts, including contacts **162***a*, **162***b*, for connection into a corresponding connector in the handle of the scanner **155**. FIGS. **9** and **11** illustrate the second end connector **164** as a plug connector (similar to the type conventionally used on home telephones) having a plurality of wire connectors (including wire connectors **164***a*, **164***b*) for connection into a corresponding connector in the data terminal **180**.

FIG. **12** schematically illustrates an example cable connection scheme **190** for the cable connector **160** of FIG. **9**. Color coded cable wires (Brown, Orange, Black, Yellow, Green, White) provide desired electrical communication path between edge contacts (nos. **10**, **9**, **18** etc.) in the first end connector **162** and respective contacts (nos. **1**, **2**, **3**, **4**) in the second end connector **164**. The cable connector **160** also includes peripheral configuration selection by electrical connector **161** (connecting edge contacts **14** and **13**) and by electrical connector **163** (connecting edge contacts **1** and **17**).

In the preferred embodiment of the present invention as shown in FIG. **14**, the data reader device **310** is further (or alternately) configured by reading a label having encoded data thereon such as a configuration bar code **335** which is placed on the interconnect cable **320**. By reading the configuration code **335**, the scanner **310** (and/or the host **340**) is configured for the particular application on the basis of the encoded instruction data by means of an internal configuration routine within the scanner **310** or the host **340**.

U.S. Pat. Nos. 4,866,257 and 4,861,972 (herein incorporated by reference) disclose examples on how a scanner may be configured by scanning a bar code or by downloading information from a host computer. Once the bar code has been scanned or the control information, the configuration information is stored in a memory (preferably a non-volatile memory such as EEPROM) in the scanner so that repetitive configuration is not required and the configuration of the scanner is not lost when power is turned off. Typically, the bar codes are contained in the user manual and the user must obtain the manual and then select the correct bar code to be scanned. However, according to the preferred embodiment shown in FIG. **14**, the user **305** need only select the correct interconnect cable **320** and the correct bar code label **335** to be scanned is automatically selected since it is on the cable itself. Moreover, being on the cable, the bar code is readily accessible and locatable without having to locate the user manual. Though once the system has been initially configured, the system will preferably store the configuration information, if it becomes necessary to reconfigure, the configuration bar code **335** remains readily accessible on the cable **320**.

In order to ensure that a scanner is not inadvertently reconfigured, a configuration switch may be provided which must be actuated to place the scanner in programming mode. Once the switch is actuated the scanner enters programming mode enabling the programming label **335** on the cable to be read and configure the scanner. The switch may be a hardware switch such as a dip switch **312** (see FIG. **14**) on the housing of the scanner **310**. Alternately, programming mode may be entered (and exited if desired) by a soft switch such as a switch label **337** located on the cable **320**. Though the configuring bar code conveniently appears on the interconnect cable, configuration bar codes (i.e., the same bar code as appearing on the cable as well as additional bar codes) may nonetheless be provided in the user manual to allow the user to configure the peripheral as desired such as to specific user optional settings. Alternately, the programming label may be positioned on the host computer or terminal. Such a location may be less desirable since different scanners may be plugged into the same host, possibly requiring different programming labels. Alternately, the program switch label may be positioned on the host as shown by the label **342** on the host **340** of FIG. **14**. The user would then scan the label **342** to switch to programming mode and then scan the programming label **335** on the cable **320**.

Configuration on the basis of the encoded data instructions obtained by reading the label or bar code **335** may be accomplished by a suitable internal configuration routine. For example, the routine may configure by way of selecting proper internal switch settings or by selecting and running a given protocol program. The data reader preferably stores its configuration parameters in a non-volatile programmable memory such as EEPROM. These parameters may be set by manual programming or reset by the configuration routine. The instructions from the label may cause the software in the data reader to execute a series of commands resulting in the setting (or resetting) of the EEPROM-stored parameters.

Configuration selection or parameters change or set a particular function for the peripheral. Possible configuration selections or parameters may include, by way of example for a handheld scanner:

- interface identification (for a laser scanner, may include Undecoded, IBM 4683, OCIA, RS-232, Wand Emulation, etc.);
- communication parameters such as baud rate (2400 baud, 9600 baud etc.); data format settings (parity, stop bits, data bits), hardware handshaking (CTS/RTS), software handshaking (Xon,Xoff), intercharacter delay (none, 10 ms, 20 ms etc.), UPC Data Format (UPA-A, UPC-E, Check Digit, Number System Digit);
- system specific parameters (prefixes, suffixes, symbology identifiers, etc.);
- reading restrictions, the instructions might restrict the set of codes options that the decoder may have to handle; by restricting the reading options, the operation speed

of the "autodiscrimination" algorithm (the means by which the decoder figures out which code it is seeing) may be increased as compared to requiring the algorithm to consider all code types.

The preferred actual location on the cable for the encoded label will depend upon the particular application.

Referring to FIG. 5 for example, the label **135** is located adjacent the second end connector. An alternate location is directly on the second end connector **124** as shown by symbol label **133** on end connector **124**. If the second end connector **124** is too small to practically accommodate the label **133** or if the connection location to the data terminal **140** provides inconvenient access, the label **135** may preferably be located up the cable connector **120** at a suitable distance from the second end connector **124**. In general, it is impractical to locate the label **135** on the first end connector **122** or immediately adjacent thereto because when the first end connector **122** is plugged into the scanner **155**, the scanner **155** cannot be oriented to scan a label located immediately adjacent the first end connector **122**.

Nonetheless, in certain applications it may be desirable to locate the label **135** near the first end connector **122**. If the connector cable **120** is relatively long, for example 50 feet (15 meters), the label may be preferably positioned about 2 feet (60 cm) from the first end connector **122**. Such a position is close enough to be easily located but far enough to allow convenient access. Such a location is illustrated in FIG. **14** where the bar code label **335** is located on the cable **320** at a convenient distance from scanner **310**.

Though the cable connector configuration embodiment and the cable connector data reader configuration embodiment may be used separately to configure the peripheral, the embodiments may be combined together to provide a comprehensive and readily implemented configuration procedure. By way of example, FIG. **15** illustrates a peripheral shown as a laser scanner **410** is equipped with a multi-interface architecture which allows any one of many different types of host interfaces (such as a data terminal **440**, a fixed scanner or other point of sale unit **450**, or a handheld key entry unit **460**) to be selected for a particular scanner. A cable **420** is selected corresponding to the particular scanner **410** and the desired host interface (**440, 450, 460**) pair. In practice, the user is supplied with a plurality of specific connector cables which correspond to the possible scanner/host pairs which the customer may have.

When plugged into the scanner, the cable **420** itself selects a certain configuration scheme in the scanner **410**. The cable **420** may also configure the host interface. The cable **420** may be equipped with physical connector elements to ensure that only the correct cable may be even physically plugged into the host. Once plugged into the scanner and the host, using the scanner itself, the operator then scans the label on the cable which completes or confirms scanner/host configuration.

Besides the bar code label, there are other types of labels or tags containing information which may be obtained by a reader device. One such label or tag is an RFID tag (radio frequency identification tag). The RFID tag is normally passive, but when activated or prompted by a signal from a interrogator, the RFID tag emits a signal with its information to a receiving device. In one alternative embodiment, the cable **320** may include a label **336** comprising an RFID tag instead of a bar code. The scanner **310** may comprise an RFID tag interrogator/receiver (either exclusively or in combination with a bar code scanning mechanism) which prompts the RFID tag **336** for its data and receives the data. The data is then used to set internal configuration, function or the like. The programming RFID tag process may be initiated for example by actuating the programming switch **312** on the device **310**. Alternately, programming may be automatically initiated upon power up or power down of the device **310**. In the power up example, when the device is powered up, the programming frequency signal would be emitted and upon receipt the RFID tag would emit its signal transmitting programming data to the device. Such a system would not require the user to perform any act except plug in the correct cable and turn on the data reader and the data reader itself would obtain the proper configuration information (for example) from the RFID tag on the cable and in this example configure itself for the host device such as by selecting the proper internal parameter.

An RFID programming tag **343** may alternately be located on the host device itself. By activating the programming sequence, the device **310** may receive the data from the tag **343** identifying the host device **340** allowing the peripheral device to be configured for that host device.

Thus, a peripheral configuration system and method have been shown and described. Though certain examples and advantages have been disclosed, further advantages and modifications may become obvious to one skilled in the art from the disclosures herein. The invention therefore is not to be limited except in the spirit of the claims that follow.

We claim:

1. A data reading system comprising:

a handheld terminal;

a translator module plugged into the terminal;

a cable plugged into the translator module;

a handheld data reader connected to the translator module via the cable,

wherein the translator module interfaces the data reader to the handheld terminal.

2. A data reading system according to claim 1 further comprising

a central processing unit;

a communications link for providing communication between the handheld terminal and the central processing unit.

3. A data reading system according to claim **1** wherein the data reader is selected from the group consisting of: a bar code scanner, a CCD reader, and an RFID tag reader.

4. A data reading system according to claim **1** wherein the cable comprises a printed circuit board equipped with switches which may be set for changing operation of the cable.

5. A portable data reading system comprising:

a handheld terminal having a housing including an externally accessible connector;

a translator module removably plugged into the connector on the handheld terminal;

a handheld data reader connected to the handheld terminal via the translator module, the data reader reading an optical code and generating an electrical signal corresponding thereto for transmitting to the handheld terminal,

wherein the translator module converts the signal transmitted from the data reader into a signal of the same form as that produced by the handheld terminal.

6. A portable data reading system according to claim **5** wherein the data reader is selected from the group consisting of: a bar code scanner, a CCD reader, and an RFID tag reader.

7. A portable data reading system according to claim **5** further comprising a cable for connecting the data reader to the handheld terminal.

8. A portable data reading system according to claim 5 wherein the cable has a first end plugged into the translator module and a second end plugged into the data reader.

9. A portable data reading system according to claim 5 further comprising a bar code label containing configuration data, wherein the data reader is laser bar code scanner, wherein scanner obtains the configuration data by scanning the bar code label and uses the configuration data to change a function of the scanner.

10. A method of data reading comprising the steps of:

providing a handheld terminal with an externally accessible connector;

plugging a translator module into the connector on the handheld terminal;

connecting a portable data reader to the translator module, the translator module converting a signal from the data reader into a form accepted by the handheld terminal.

11. A method according to claim 10 further comprising the step of interfacing the data reader to the handheld terminal via the translator module.

12. A method according to claim 10 further comprising the step of

activating the data reader and the handheld terminal;

obtaining the data from the label with the data reader;

setting a function of the data reading system on the basis of the data obtained from the label.

13. A method according to claim 12 wherein the label comprises a bar code label and the data reader comprises a barcode reader.

14. A method according to claim 12 wherein the label comprises a radio frequency identification tag and the data reader comprises a radio frequency identification reader.

15. A method according to claim 10 wherein the data reader comprises a handheld laser scanner.

16. A method according to claim 10 further comprising connecting the data reader to the translator module via a cable.

* * * * *